Example 1.1

1. Two students witness burglar Bill making off from a crime scene in his getaway car. The first student tells the police that the number plate began with and R or a P and that the first numerical digit was either a 2 or a 3. The second student recalls that the last letter was an M or an N. Given that all number plates have the same format: two capital letters (between A and Z) followed by 2 digits (between 0 and 9) followed by 3 more letters, how many number plates must the police investigate.

2. There are 7 people to be seated at a round table. How many seating arrangements are possible? How many times must they change places so that everyone sits next to everyone else at least once. What difference does in make if one person always sits in the same place?
3. A lecturer divides a class of 30 students into 5 groups, not necessarily of the same size, and then chooses one representative from each group. In how many ways is this possible? If some of the groups are to be selected to move into another room how many possibilities are there now?

2. There are 7 people to be seated at a round table. How many seating arrangements are possible? How many times must they change places so that everyone sits next to everyone else at least once. What difference does in make if one person always sits in the same place?
3. A lecturer divides a class of 30 students into 5 groups, not necessarily of the same size, and then chooses one representative from each group. In how many ways is this possible? If some of the groups are to be selected to move into another room how many possibilities are there now?

A basic counting technique

I have shirts of 3 different colours, trousers of 2 different colours and socks of 5 different colours. How many different outfits (colour combinations) are available to me?

The general rule is

Lemma 1.2

A task is to be carried out in stages. There are n_1 ways of carrying out the first stage. For each of these there are n_2 ways of carrying out second stage. For each of these n_2 ways there are n_3 ways of doing the third stage and so on. If there are r stages then there are in total $n_1 n_2 \cdots n_r$ ways of carrying out the entire task.

A basic counting technique

I have shirts of 3 different colours, trousers of 2 different colours and socks of 5 different colours. How many different outfits (colour combinations) are available to me?

The general rule is

Lemma 1.2

A task is to be carried out in stages. There are n_1 ways of carrying out the first stage. For each of these there are n_2 ways of carrying out second stage. For each of these n_2 ways there are n_3 ways of doing the third stage and so on. If there are r stages then there are in total $n_1 n_2 \cdots n_r$ ways of carrying out the entire task.

The Pigeonhole Principle

Do any 2 Newcastle students share the same Personal Identification Number (PIN) for their debit cards? Is your PIN number the same as mine?

More generally we have the following lemma.

Lemma 1.3

If n identical balls are put into k boxes and $n > k$ then some box contains at least 2 balls.

The Pigeonhole Principle

Do any 2 Newcastle students share the same Personal Identification Number (PIN) for their debit cards? Is your PIN number the same as mine?

More generally we have the following lemma.

Lemma 1.3

If n identical balls are put into k boxes and $n > k$ then some box contains at least 2 balls.

Coming back to the question of PINs we can worry even more. Is it possible that 3 or more people in Newcastle share the same PIN?

Lemma 1.4

Suppose n identical balls are placed in k boxes and that $n > kr$, for some positive integer r. Then some box contains at least $r + 1$ balls.

Coming back to the question of PINs we can worry even more. Is it possible that 3 or more people in Newcastle share the same PIN?

Lemma 1.4

Suppose n identical balls are placed in k boxes and that $n > kr$, for some positive integer r. Then some box contains at least $r + 1$ balls.

Example 1.5

Suppose that 7 boys dance with 7 girls, all on the dance floor at once. How many pairings are possible?

Definition 1.6

A map $f : X \to Y$ is called

1. an injection if $a \neq b$ implies $f(a) \neq f(b)$, for all $a, b \in X$;
2. a surjection if, for all $y \in Y$, there is $x \in X$ with $f(x) = y$;
3. a bijection if f is an injection and a surjection.

Also one-one means the same as injection and onto means the same as surjection.

Example 1.5

Suppose that 7 boys dance with 7 girls, all on the dance floor at once. How many pairings are possible?

Definition 1.6

A map $f : X \to Y$ is called

1. an injection if $a \neq b$ implies $f(a) \neq f(b)$, for all $a, b \in X$;
2. a surjection if, for all $y \in Y$, there is $x \in X$ with $f(x) = y$;
3. a bijection if f is an injection and a surjection.

Also one-one means the same as injection and onto means the same as surjection.

Definition 1.7

A bijection from a set X to itself is called a permutation.

Theorem 1.8

The number of permutations of a set of n elements is $n!$.

Definition 1.7

A bijection from a set X to itself is called a permutation.

Theorem 1.8

The number of permutations of a set of n elements is $n!$.

In the examples of permutations above each element of the set $\{1, \ldots, n\}$ appears exactly once.

By contrast suppose that I drink 5 cups of water, 3 cups of tea and 2 cups of coffee every day. How many different ways can I arrange the order in which I drink all these drinks?

A multiset is a collection of elements of a set in which elements may occur more than once.

In the examples of permutations above each element of the set $\{1, \ldots, n\}$ appears exactly once.

By contrast suppose that I drink 5 cups of water, 3 cups of tea and 2 cups of coffee every day. How many different ways can I arrange the order in which I drink all these drinks?

A multiset is a collection of elements of a set in which elements may occur more than once.

In the examples of permutations above each element of the set $\{1, \ldots, n\}$ appears exactly once.

By contrast suppose that I drink 5 cups of water, 3 cups of tea and 2 cups of coffee every day. How many different ways can I arrange the order in which I drink all these drinks?

A multiset is a collection of elements of a set in which elements may occur more than once.

Theorem 1.9

Let $a_1, \ldots, a_k$ be positive integers and let $n = a_1 + \cdots + a_k$.

If we have a multiset of a_1 elements of type 1,

a_2 elements of type 2,

... , a_k elements of type k,

then we can arrange these elements in order in

$$\frac{n!}{a_1! \cdots a_k!}$$

ways.

Theorem 1.9

Let $a_1, \ldots, a_k$ *be positive integers and let* $n = a_1 + \cdots + a_k$.

If we have a multiset of a_1 *elements of type* 1,

a_2 *elements of type* 2,

... , a_k *elements of type* k,

then we can arrange these elements in order in

$$\frac{n!}{a_1! \cdots a_k!}$$

ways.

Theorem 1.9

Let $a_1, \ldots, a_k$ be positive integers and let $n = a_1 + \cdots + a_k$.

If we have a multiset of a_1 elements of type 1,

a_2 elements of type 2,

... , a_k elements of type k,

then we can arrange these elements in order in

$$\frac{n!}{a_1! \cdots a_k!}$$

ways.

A database stores information of a certain type as a string of length 10 consisting of capital letters A–Z, lower case letters a–z and numerical digits 0–9: so there are 62 different symbols available. How many different records can be made this way?

Lemma 1.10

The number of sequences $a_1, \ldots, a_k$ of length k where all the elements a_i belong to a set of size n is n^k.

A database stores information of a certain type as a string of length 10 consisting of capital letters A–Z, lower case letters a–z and numerical digits 0–9: so there are 62 different symbols available. How many different records can be made this way?

Lemma 1.10

The number of sequences $a_1, \ldots, a_k$ of length k where all the elements a_i belong to a set of size n is n^k.

Example 1.11

I am going to paint each of my fingernails (and thumbnails) a different colour. I have paints of 5 different colours. How many different ways can I do this?

Corollary 1.12

The number of maps from a set of size k to a set of size n is n^k.

I'm going to place a bet at Cheltenham races on a race in which there are 15 horses. Only the order of the first 3 horses over the line is recorded and I bet that horses Brave Inca, Straw Bear and Lazy Champion will come in 1st, 2nd and 3rd, respectively. How many outcomes are possible and how likely am I to win my bet?

Theorem 1.13

The number of ordered k-subsets of a set of size n is

$$n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}.$$

I'm going to place a bet at Cheltenham races on a race in which there are 15 horses. Only the order of the first 3 horses over the line is recorded and I bet that horses Brave Inca, Straw Bear and Lazy Champion will come in 1st, 2nd and 3rd, respectively. How many outcomes are possible and how likely am I to win my bet?

Theorem 1.13

The number of ordered k-subsets of a set of size n is

$$n(n-1)\cdots(n-k+1) = \frac{n!}{(n-k)!}.$$

Example 1.14

How many subsets does the set $\{a, b, c\}$ have?

Lemma 1.15

The number of subsets of a set of size n is 2^n.

Example 1.14

How many subsets does the set $\{a, b, c\}$ have?

Lemma 1.15

The number of subsets of a set of size n is 2^n.

k-subsets

In the Quickfire Lotto game players buy a ticket and select 4 numbers from a list of the numbers from 1 to 48. Then 4 different winning numbers between 1 and 48 are selected at random. How many tickets would you need to buy to be sure of getting all 4 numbers (and so winning the top prize).

By convention we set $0! = 1$.

Definition 1.16

The binomial coefficient for integers $n \geq k \geq 0$ is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

For integers $n < k$ we define

$$\binom{n}{k} = 0.$$

In particular from this definition we have

$$\binom{n}{0} = \binom{n}{n} = \binom{0}{0} = 1 \text{ and } \binom{n}{1} = \binom{n}{n-1} = n.$$

By convention we set $0! = 1$.

Definition 1.16

The binomial coefficient for integers $n \geq k \geq 0$ is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

For integers $n < k$ we define

$$\binom{n}{k} = 0.$$

In particular from this definition we have

$$\binom{n}{0} = \binom{n}{n} = \binom{0}{0} = 1 \text{ and } \binom{n}{1} = \binom{n}{n-1} = n.$$

By convention we set $0! = 1$.

Definition 1.16

The binomial coefficient for integers $n \geq k \geq 0$ is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

For integers $n < k$ we define

$$\binom{n}{k} = 0.$$

In particular from this definition we have

$$\binom{n}{0} = \binom{n}{n} = \binom{0}{0} = 1 \text{ and } \binom{n}{1} = \binom{n}{n-1} = n.$$

By convention we set $0! = 1$.

Definition 1.16

The binomial coefficient for integers $n \geq k \geq 0$ is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

For integers $n < k$ we define

$$\binom{n}{k} = 0.$$

In particular from this definition we have

$$\binom{n}{0} = \binom{n}{n} = \binom{0}{0} = 1 \text{ and } \binom{n}{1} = \binom{n}{n-1} = n.$$

Theorem 1.17

The number of k-subsets of a set of n elements is

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k!}.$$

Example 1.18

It has been decided that classes for module MAS9999 will all be held on a Friday between the hours of 8:00 and 20:00 (so there are 12 hour long slots available). There are to be 5 hours of teaching but no two consecutive hours. In how many ways can the schedule be devised?

Theorem 1.17

The number of k-subsets of a set of n elements is

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k+1)}{k!}.$$

Example 1.18

It has been decided that classes for module MAS9999 will all be held on a Friday between the hours of 8:00 and 20:00 (so there are 12 hour long slots available). There are to be 5 hours of teaching but no two consecutive hours. In how many ways can the schedule be devised?

k-multisets

Example

Suppose that, in the Quickfire Lotto game described above, instead of choosing 4 different numbers from the list $1, \ldots, 48$ we choose any 4 such numbers with repetition: that is we choose a multiset of 4 elements. I win if my numbers are the same as a 4-multiset chosen from $1, \ldots, 48$ at random by the lottery company. How many tickets do I need to buy to ensure I win this game?

A collection of k elements of a set where repetition is allowed is called a k-multiset.

Theorem 1.19

The number of k-multisets of a set of n elements is

$$\binom{n+k-1}{k}.$$

k-multisets

Example

Suppose that, in the Quickfire Lotto game described above, instead of choosing 4 different numbers from the list $1, \dots, 48$ we choose any 4 such numbers with repetition: that is we choose a multiset of 4 elements. I win if my numbers are the same as a 4-multiset chosen from $1, \dots, 48$ at random by the lottery company. How many tickets do I need to buy to ensure I win this game?

A collection of k elements of a set where repetition is allowed is called a k-multiset.

Theorem 1.19

The number of k-multisets of a set of n elements is

$$\binom{n+k-1}{k}.$$

k-multisets

Example

Suppose that, in the Quickfire Lotto game described above, instead of choosing 4 different numbers from the list $1, \ldots, 48$ we choose any 4 such numbers with repetition: that is we choose a multiset of 4 elements. I win if my numbers are the same as a 4-multiset chosen from $1, \ldots, 48$ at random by the lottery company. How many tickets do I need to buy to ensure I win this game?

A collection of k elements of a set where repetition is allowed is called a k-multiset.

Theorem 1.19

The number of k-multisets of a set of n elements is

$$\binom{n+k-1}{k}.$$

The Binomial Theorem

Consider expanding

$$(x+y)^7 = (x+y)(x+y)(x+y)(x+y)(x+y)(x+y)(x+y).$$

What is the coefficient of say x^3y^4 in the result?

Theorem 1.20 (The Binomial Theorem)

For all positive integers n

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}.$$

The Binomial Theorem

Consider expanding

$$(x+y)^7 = (x+y)(x+y)(x+y)(x+y)(x+y)(x+y)(x+y).$$

What is the coefficient of say x^3y^4 in the result?

Theorem 1.20 (The Binomial Theorem)

For all positive integers n

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}.$$

The Binomial Theorem

Consider expanding

$$(x+y)^7 = (x+y)(x+y)(x+y)(x+y)(x+y)(x+y)(x+y).$$

What is the coefficient of say x^3y^4 in the result?

Theorem 1.20 (The Binomial Theorem)

For all positive integers n

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}.$$

Corollary 1.21

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

and

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0.$$

In long hand:

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n-1} + \binom{n}{n} = 2^n$$

and

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^{n-1}\binom{n}{n-1} + (-1)^n \binom{n}{n} = 0.$$

Corollary 1.21

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

and

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0.$$

In long hand:

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n-1} + \binom{n}{n} = 2^n$$

and

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^{n-1}\binom{n}{n-1} + (-1)^n \binom{n}{n} = 0.$$

Corollary 1.21

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

and

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0.$$

In long hand:

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n-1} + \binom{n}{n} = 2^n$$

and

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^{n-1}\binom{n}{n-1} + (-1)^n \binom{n}{n} = 0.$$

Corollary 1.21

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

and

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0.$$

In long hand:

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{n-1} + \binom{n}{n} = 2^n$$

and

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \cdots + (-1)^{n-1}\binom{n}{n-1} + (-1)^n \binom{n}{n} = 0.$$

Lemma 1.22

Let n and k be positive integers.

(i)

$$\binom{n}{k} = \binom{n}{n-k}.$$

(ii)

$$\binom{n-1}{k} + \binom{n-1}{k-1} = \binom{n}{k}.$$

Lemma 1.22

Let n and k be positive integers.

(i)

$$\binom{n}{k} = \binom{n}{n-k}.$$

(ii)

$$\binom{n-1}{k} + \binom{n-1}{k-1} = \binom{n}{k}.$$

The Multinomial Theorem

Suppose we wish to compute powers of $(x + y + z)$ instead of $(x + y)$.

For example we have

$$(x+y+z)^3 = x^3 + y^3 + z^3 \\ + 3x^2y + 3x^2z + 3xy^2 + 3y^2z + 3xz^2 + 3yz^2 \\ + 6xyz.$$

What is the coefficient of $x^a y^b z^c$ in $(x + y + z)^n$?

The Multinomial Theorem

Suppose we wish to compute powers of $(x + y + z)$ instead of $(x + y)$.

For example we have

$$\begin{aligned}(x+y+z)^3 &= x^3 + y^3 + z^3 \\ &+ 3x^2y + 3x^2z + 3xy^2 + 3y^2z + 3xz^2 + 3yz^2 \\ &+ 6xyz.\end{aligned}$$

What is the coefficient of $x^a y^b z^c$ in $(x + y + z)^n$?

The Multinomial Theorem

Suppose we wish to compute powers of $(x+y+z)$ instead of $(x+y)$.

For example we have

$$(x+y+z)^3 = x^3+y^3+z^3 \\ +3x^2y+3x^2z+3xy^2+3y^2z+3xz^2+3yz^2 \\ +6xyz.$$

What is the coefficient of $x^ay^bz^c$ in $(x+y+z)^n$?

Definition 1.23

Let $n = a_1 + \cdots + a_k$, where a_i is a non-negative integer, $i = 1, \ldots, k$.

Define the multinomial coefficient

$$\binom{n}{a_1, \ldots, a_k} = \frac{n!}{a_1! \cdots a_k!}.$$

Theorem 1.24

Let $x_1, \ldots, x_k$ be real numbers. Then, for all non-negative integers n and positive integers k, we have

$$(x_1 + \cdots + x_k)^n = \sum_{a_1, \ldots, a_k} \binom{n}{a_1, \ldots, a_k} x_1^{a_1} \cdots x_k^{a_k},$$

where the sum is over all length k sequences $a_1, \ldots, a_k$ of non-negative integers such that $n = a_1 + \cdots + a_k$.

Definition 1.23

Let $n = a_1 + \cdots + a_k$, where a_i is a non-negative integer, $i = 1, \ldots, k$.

Define the multinomial coefficient

$$\binom{n}{a_1, \ldots, a_k} = \frac{n!}{a_1! \cdots a_k!}.$$

Theorem 1.24

Let $x_1, \ldots, x_k$ be real numbers. Then, for all non-negative integers n and positive integers k, we have

$$(x_1 + \cdots + x_k)^n = \sum_{a_1, \ldots, a_k} \binom{n}{a_1, \ldots, a_k} x_1^{a_1} \cdots x_k^{a_k},$$

where the sum is over all length k sequences $a_1, \ldots, a_k$ of non-negative integers such that $n = a_1 + \cdots + a_k$.

Definition 1.23

Let $n = a_1 + \cdots + a_k$, where a_i is a non-negative integer, $i = 1, \ldots, k$.

Define the multinomial coefficient

$$\binom{n}{a_1, \ldots, a_k} = \frac{n!}{a_1! \cdots a_k!}.$$

Theorem 1.24

Let $x_1, \ldots, x_k$ be real numbers. Then, for all non-negative integers n and positive integers k, we have

$$(x_1 + \cdots + x_k)^n = \sum_{a_1, \ldots, a_k} \binom{n}{a_1, \ldots, a_k} x_1^{a_1} \cdots x_k^{a_k},$$

where the sum is over all length k sequences $a_1, \ldots, a_k$ of non-negative integers such that $n = a_1 + \cdots + a_k$.

Inclusion-Exclusion

I have 30 uncles some wicked some virtuous. 12 of them smoke, 12 of them drink and 18 of them gamble. 6 smoke and drink, 9 drink and gamble, 8 smoke and gamble and finally 5 smoke, drink and gamble. How many neither smoke, drink nor gamble?

The general result covering the example above is the next theorem.

Inclusion-Exclusion

I have 30 uncles some wicked some virtuous. 12 of them smoke, 12 of them drink and 18 of them gamble. 6 smoke and drink, 9 drink and gamble, 8 smoke and gamble and finally 5 smoke, drink and gamble. How many neither smoke, drink nor gamble?

The general result covering the example above is the next theorem.

Inclusion-Exclusion

Theorem 1.25

Let $A_1, \ldots, A_k$ *be subsets of a set* E*. Then*

$$
\begin{aligned}
|A_1 \cup \cdots \cup A_k| = {} & |A_1| + \cdots + |A_k| \\
& - (|A_1 \cap A_2| + \cdots + |A_{k-1} \cap A_k|) \\
& + (|A_1 \cap A_2 \cap A_3| + \cdots + |A_{k-2} \cap A_{k-1} \cap A_k|) \\
& \vdots \\
& + (-1)^{k-1} |A_1 \cap \cdots \cap A_k|.
\end{aligned}
$$

That is

$$|A_1 \cup \cdots \cup A_k| = \sum_{i=1}^{k} (-1)^{i-1} \sum_{s_1,\ldots,s_i} |A_{s_1} \cap \cdots \cap A_{s_i}|$$

where, for all i*, the subscripts* $s_1, \ldots, s_i$ *run over all* i*-subsets of* $\{1, \ldots, k\}$.

Inclusion-Exclusion

Theorem 1.25

Let $A_1, \dots, A_k$ be subsets of a set E. Then

$$
\begin{aligned}
|A_1 \cup \cdots \cup A_k| = |A_1| + \cdots + |A_k| \\
&- (|A_1 \cap A_2| + \cdots + |A_{k-1} \cap A_k|) \\
&+ (|A_1 \cap A_2 \cap A_3| + \cdots + |A_{k-2} \cap A_{k-1} \cap A_k|) \\
&\vdots \\
&+ (-1)^{k-1} |A_1 \cap \cdots \cap A_k|.
\end{aligned}
$$

That is

$$|A_1 \cup \cdots \cup A_k| = \sum_{i=1}^{k} (-1)^{i-1} \sum_{s_1, \dots, s_i} |A_{s_1} \cap \cdots \cap A_{s_i}|$$

where, for all i, the subscripts $s_1, \dots, s_i$ run over all i-subsets of $\{1, \dots, k\}$.

Derangements

Example 1.26

n people come to a party at your house, each wearing a hat. When they leave they are not so sober and they can't remember which hat is which. In the morning each person discovers they have someone else's hat. How many ways can this happen?

A permutation with no fixed points is called a derangement of a setand the number of such permutations of an n-set is denoted $D(n)$.

Theorem 1.27

The number of derangements of an n-set is

$$D(n) = \sum_{r=0}^{n} (-1)^r \frac{n!}{r!}.$$

Derangements

Example 1.26

n people come to a party at your house, each wearing a hat. When they leave they are not so sober and they can't remember which hat is which. In the morning each person discovers they have someone else's hat. How many ways can this happen?

A permutation with no fixed points is called a derangement of a setand the number of such permutations of an n-set is denoted $D(n)$.

Theorem 1.27

The number of derangements of an n-set is

$$D(n) = \sum_{r=0}^{n} (-1)^r \frac{n!}{r!}.$$

Derangements

Example 1.26

n people come to a party at your house, each wearing a hat. When they leave they are not so sober and they can't remember which hat is which. In the morning each person discovers they have someone else's hat. How many ways can this happen?

A permutation with no fixed points is called a derangement of a setand the number of such permutations of an n-set is denoted $D(n)$.

Theorem 1.27

The number of derangements of an n-set is

$$D(n) = \sum_{r=0}^{n} (-1)^r \frac{n!}{r!}.$$

Compositions

Suppose I wish to distribute n toffees to k students. How many ways is it possible to do this?

Now suppose that I feel bad about the possibility that some students may not get any toffees atall. How many ways are there of distributing n toffees amongst k students so that every student gets at least one toffee.

Compositions

Suppose I wish to distribute n toffees to k students. How many ways is it possible to do this?

Now suppose that I feel bad about the possibility that some students may not get any toffees atall. How many ways are there of distributing n toffees amongst k students so that every student gets at least one toffee.

More formally we make the following definition.

Definition 1.28

A sequence $(a_1, \ldots, a_k)$ of k non-negative integers such that $\sum_{i=1}^{k} a_i = n$ is called a weak composition of n into k parts. If $a_i > 0$ for all i the sequence is called a composition.

Theorem 1.29

The number of weak compositions of n into k parts is

$$\binom{n+k-1}{n} = \binom{n+k-1}{k-1}.$$

Corollary 1.30

The number of compositions of n into k parts is

$$\binom{n-1}{k-1}.$$

More formally we make the following definition.

Definition 1.28

A sequence $(a_1, \ldots, a_k)$ of k non-negative integers such that $\sum_{i=1}^{k} a_i = n$ is called a weak composition of n into k parts. If $a_i > 0$ for all i the sequence is called a composition.

Theorem 1.29

The number of weak compositions of n into k parts is

$$\binom{n+k-1}{n} = \binom{n+k-1}{k-1}.$$

Corollary 1.30

The number of compositions of n into k parts is

$$\binom{n-1}{k-1}.$$

More formally we make the following definition.

Definition 1.28

A sequence $(a_1, \ldots, a_k)$ of k non-negative integers such that $\sum_{i=1}^{k} a_i = n$ is called a weak composition of n into k parts. If $a_i > 0$ for all i the sequence is called a composition.

Theorem 1.29

The number of weak compositions of n into k parts is

$$\binom{n+k-1}{n} = \binom{n+k-1}{k-1}.$$

Corollary 1.30

The number of compositions of n into k parts is

$$\binom{n-1}{k-1}.$$

Since a_i can be zero in a weak composition there exist weak compositions of n into k parts for all $k > 0$.

However, for a composition of n into k parts to exist we must have $k \leq n$.

Therefore there are finitely many compositions of n: and we have the following Corollary.

Corollary 1.31

The number of compositions of n is 2^{n-1}.

Since a_i can be zero in a weak composition there exist weak compositions of n into k parts for all $k > 0$.

However, for a composition of n into k parts to exist we must have $k \leq n$.

Therefore there are finitely many compositions of n: and we have the following Corollary.

Corollary 1.31

The number of compositions of n is 2^{n-1}.

Since a_i can be zero in a weak composition there exist weak compositions of n into k parts for all $k > 0$.

However, for a composition of n into k parts to exist we must have $k \leq n$.

Therefore there are finitely many compositions of n: and we have the following Corollary.

Corollary 1.31

The number of compositions of n is 2^{n-1}.

Since a_i can be zero in a weak composition there exist weak compositions of n into k parts for all $k > 0$.

However, for a composition of n into k parts to exist we must have $k \leq n$.

Therefore there are finitely many compositions of n: and we have the following Corollary.

Corollary 1.31

The number of compositions of n *is* 2^{n-1}.

Partitions

Suppose now I have n flowers, each one a different type, and I wish to arrange them in k different vases, in such a way that there is at least one flower in each vase.

In how many ways can I do this?

Partitions

Suppose now I have n flowers, each one a different type, and I wish to arrange them in k different vases, in such a way that there is at least one flower in each vase.

In how many ways can I do this?

Stirling numbers

Definition 1.32

A partition of a set X into k parts is a collection $S_1, \ldots, S_k$ of non-empty subsets of X such that $X = \cup_{i=1}^{k} S_i$ and $S_i \cap S_j = \emptyset$, whenever $i \neq j$.

Example 1.33

List all the partitions of the set $\{1, 2, 3, 4\}$ into 2 non-empty subsets.

Definition 1.34

The number of partitions of $\{1, \ldots, n\}$ into k parts is denoted $S(n, k)$. The numbers $S(n, k)$ are called the Stirling numbers (of the second kind).

Stirling numbers

Definition 1.32

A partition of a set X into k parts is a collection $S_1, \ldots, S_k$ of non-empty subsets of X such that $X = \cup_{i=1}^{k} S_i$ and $S_i \cap S_j = \emptyset$, whenever $i \neq j$.

Example 1.33

List all the partitions of the set $\{1, 2, 3, 4\}$ into 2 non-empty subsets.

Definition 1.34

The number of partitions of $\{1, \ldots, n\}$ into k parts is denoted $S(n, k)$. The numbers $S(n, k)$ are called the Stirling numbers (of the second kind).

Stirling numbers

Definition 1.32

A partition of a set X into k parts is a collection $S_1, \ldots, S_k$ of non-empty subsets of X such that $X = \cup_{i=1}^{k} S_i$ and $S_i \cap S_j = \emptyset$, whenever $i \neq j$.

Example 1.33

List all the partitions of the set $\{1, 2, 3, 4\}$ into 2 non-empty subsets.

Definition 1.34

The number of partitions of $\{1, \ldots, n\}$ into k parts is denoted $S(n, k)$. The numbers $S(n, k)$ are called the Stirling numbers (of the second kind).

From the example above we have $S(4,2) = 7$.

Example 1.35

Find $S(3,1)$ and $S(3,2)$.

Lemma 1.36

Let $1 \leq k \leq n$. Then

$$S(n,n) = 1 \text{ and } S(n,1) = 1,$$

$$S(n,n-1) = \binom{n}{2} \text{ and}$$

$$S(n,k) = S(n-1,k-1) + kS(n-1,k).$$

From the example above we have $S(4,2) = 7$.

Example 1.35

Find $S(3,1)$ and $S(3,2)$.

Lemma 1.36

Let $1 \leq k \leq n$. Then

$$S(n,n) = 1 \text{ and } S(n,1) = 1,$$
$$S(n,n-1) = \binom{n}{2} \text{ and}$$
$$S(n,k) = S(n-1,k-1) + kS(n-1,k).$$

From the example above we have $S(4,2) = 7$.

Example 1.35

Find $S(3,1)$ and $S(3,2)$.

Lemma 1.36

Let $1 \leq k \leq n$. *Then*

$$S(n,n) = 1 \text{ and } S(n,1) = 1,$$

$$S(n,n-1) = \binom{n}{2} \text{ and}$$

$$S(n,k) = S(n-1,k-1) + kS(n-1,k).$$

From the example above we have $S(4,2) = 7$.

Example 1.35

Find $S(3,1)$ and $S(3,2)$.

Lemma 1.36

Let $1 \leq k \leq n$. Then

$$S(n,n) = 1 \text{ and } S(n,1) = 1,$$

$$S(n,n-1) = \binom{n}{2} \text{ and}$$

$$S(n,k) = S(n-1,k-1) + kS(n-1,k).$$

From the example above we have $S(4,2) = 7$.

Example 1.35

Find $S(3,1)$ and $S(3,2)$.

Lemma 1.36

Let $1 \leq k \leq n$. *Then*

$$S(n,n) = 1 \text{ and } S(n,1) = 1,$$
$$S(n,n-1) = \binom{n}{2} \text{ and}$$
$$S(n,k) = S(n-1,k-1) + kS(n-1,k).$$

Now suppose that I have arranged the n flowers in the k vases and I wish to give each vase to a different person.

How many ways can this be done? That is, how many ways are there of distributing n different types of flower amongst k people, so that each person receives at least one flower?

Theorem 1.37

The number of surjective functions from a set of size n to a set of size k is $k!S(n,k)$.

Now suppose that I have arranged the n flowers in the k vases and I wish to give each vase to a different person.

How many ways can this be done? That is, how many ways are there of distributing n different types of flower amongst k people, so that each person receives at least one flower?

Theorem 1.37

The number of surjective functions from a set of size n to a set of size k is $k!S(n,k)$.

Now suppose that I have arranged the n flowers in the k vases and I wish to give each vase to a different person.

How many ways can this be done? That is, how many ways are there of distributing n different types of flower amongst k people, so that each person receives at least one flower?

Theorem 1.37

The number of surjective functions from a set of size n to a set of size k is $k!S(n,k)$.

We can apply the inclusion-exclusion principle to obtain a formula for $S(n,k)$. The formula is not entirely satisfactory as it contains a sum of $k+1$ terms, but it is the best we can do.

Theorem 1.38

Let k and n be positive numbers. Then

$$S(n,k) = \frac{1}{k!}\sum_{d=0}^{k}(-1)^d\binom{k}{d}(k-d)^n = \sum_{d=0}^{k}(-1)^d\frac{1}{d!(k-d)!}(k-d)^n.$$

We can apply the inclusion-exclusion principle to obtain a formula for $S(n,k)$. The formula is not entirely satisfactory as it contains a sum of $k+1$ terms, but it is the best we can do.

Theorem 1.38

Let k and n be positive numbers. Then

$$S(n,k) = \frac{1}{k!}\sum_{d=0}^{k}(-1)^d\binom{k}{d}(k-d)^n = \sum_{d=0}^{k}(-1)^d\frac{1}{d!(k-d)!}(k-d)^n.$$

We can apply the inclusion-exclusion principle to obtain a formula for $S(n,k)$. The formula is not entirely satisfactory as it contains a sum of $k+1$ terms, but it is the best we can do.

Theorem 1.38

Let k and n be positive numbers. Then

$$S(n,k) = \frac{1}{k!}\sum_{d=0}^{k}(-1)^d\binom{k}{d}(k-d)^n = \sum_{d=0}^{k}(-1)^d\frac{1}{d!(k-d)!}(k-d)^n.$$

Integer Partitions

I have a bag of n identical marbles and wish to sort them into k non-empty piles, the order of which does not matter. How many ways can I do this?

Definition 1.39

Let $a_1 \geq \cdots \geq a_k \geq 1$ be integers such that $a_1 + \cdots + a_k = n$. Then $(a_, \ldots, a_k)$ is called an integer-partition of n into k parts.

The number of integer-partitions of n into k parts is denoted $p_k(n)$

and the number of all integer-partitions of n is denoted $p(n)$.

It can be shown that

$$p(n) \sim \frac{1}{4\sqrt{3}} \exp(\pi \sqrt{\frac{2n}{3}})$$

Integer Partitions

I have a bag of n identical marbles and wish to sort them into k non-empty piles, the order of which does not matter. How many ways can I do this?

Definition 1.39

Let $a_1 \geq \cdots \geq a_k \geq 1$ be integers such that $a_1 + \cdots + a_k = n$. Then $(a_, \ldots, a_k)$ is called an integer-partition of n into k parts.

The number of integer-partitions of n into k parts is denoted $p_k(n)$

and the number of all integer-partitions of n is denoted $p(n)$.

It can be shown that

$$p(n) \sim \frac{1}{4\sqrt{3}} \exp(\pi \sqrt{\frac{2n}{3}})$$

Integer Partitions

I have a bag of n identical marbles and wish to sort them into k non-empty piles, the order of which does not matter. How many ways can I do this?

Definition 1.39

Let $a_1 \geq \cdots \geq a_k \geq 1$ be integers such that $a_1 + \cdots + a_k = n$. Then $(a_, \ldots, a_k)$ is called an integer-partition of n into k parts.

The number of integer-partitions of n into k parts is denoted $p_k(n)$

and the number of all integer-partitions of n is denoted $p(n)$.

It can be shown that

$$p(n) \sim \frac{1}{4\sqrt{3}} \exp(\pi \sqrt{\frac{2n}{3}})$$

Integer Partitions

I have a bag of n identical marbles and wish to sort them into k non-empty piles, the order of which does not matter. How many ways can I do this?

Definition 1.39

Let $a_1 \geq \cdots \geq a_k \geq 1$ be integers such that $a_1 + \cdots + a_k = n$. Then $(a_, \ldots, a_k)$ is called an integer-partition of n into k parts.

The number of integer-partitions of n into k parts is denoted $p_k(n)$

and the number of all integer-partitions of n is denoted $p(n)$.

It can be shown that

$$p(n) \sim \frac{1}{4\sqrt{3}} \exp(\pi\sqrt{\frac{2n}{3}})$$

Integer Partitions

I have a bag of n identical marbles and wish to sort them into k non-empty piles, the order of which does not matter. How many ways can I do this?

Definition 1.39

Let $a_1 \geq \cdots \geq a_k \geq 1$ be integers such that $a_1 + \cdots + a_k = n$. Then $(a_{,} \ldots, a_k)$ is called an integer-partition of n into k parts.

The number of integer-partitions of n into k parts is denoted $p_k(n)$

and the number of all integer-partitions of n is denoted $p(n)$.

It can be shown that

$$p(n) \sim \frac{1}{4\sqrt{3}} \exp(\pi \sqrt{\frac{2n}{3}})$$

Summary

Permutations	
permutations of and n-set	$n!$
orderings of a_i objects of type i, where $a_1 + \cdots + a_k = n$	$\frac{n!}{a_1! \cdots a_k!}$

Sequences	
sequences of length k over an alphabet of size n or functions from a set of size k to a set of size n	n^k
ordered k-subsets of an n-set or sequences of length k without repetition	$\frac{n!}{(n-k)!}$

Subsets	
all subsets of an n-set	2^n
k-subsets of an n-set	$\binom{n}{k}$
k-multisets of an n-set	$\binom{n+k-1}{k}$

Derangements	
derangements of an n-set	$D(n) = \sum_{r=0}^{n} (-1)^r \frac{n!}{r!}$

Compositions	
weak compositions of n into k parts	$\binom{n+k-1}{k-1}$
compositions of n into k parts	$\binom{n-1}{k-1}$
compositions of n	2^{n-1}

Partitions	
partitions of an n-set into k parts	$S(n,k) = \frac{1}{k!}\sum_{d=0}^{k}(-1)^d\binom{k}{d}(k-d)^n$
surjective functions of an n-set to a k-set	$k!S(n,k)$
integer-partitions of an n-set into k parts	$p_k(n) = ??$
integer-partitions of an n-set	$p(n) \sim \frac{1}{4\sqrt{3}}\exp(\pi\sqrt{\frac{2n}{3}})$

Graph Theory

Example 2.1. 1. On arrival at a party guests shake hands with some of the people they meet in the hallway, but mostly don't shake hands after that. If we ask each person how many people they shook hands with and then add these numbers we always have an even number. Why?

Graph Theory

Example 2.1. 1. On arrival at a party guests shake hands with some of the people they meet in the hallway, but mostly don't shake hands after that. If we ask each person how many people they shook hands with and then add these numbers we always have an even number. Why?

2. Suppose there are an odd number of people at this party. If we ask each person how many other people they shook hands with then there will be an odd number of people who answer with an even number. Why?

Graph Theory

Example 2.1. 1. On arrival at a party guests shake hands with some of the people they meet in the hallway, but mostly don't shake hands after that. If we ask each person how many people they shook hands with and then add these numbers we always have an even number. Why?

2. Suppose there are an odd number of people at this party. If we ask each person how many other people they shook hands with then there will be an odd number of people who answer with an even number. Why?

3. Only 6 people make it to the MAS2216 lecture at mid-day on the day after this party. I can guarantee that either 3 of them shook each others hands or 3 of them did not. How can I be sure?

Definitions

Definition 2.2. A graph G consists of

(i) a finite non–empty set $V(G)$ of vertices and

Definitions

Definition 2.2. A graph G consists of

(i) a finite non–empty set $V(G)$ of vertices and

(ii) a set $E(G)$ of edges

Definitions

Definition 2.2. A graph G consists of

(i) a finite non–empty set $V(G)$ of vertices and

(ii) a set $E(G)$ of edges

such that every edge $e \in E(G)$ is an unordered pair $\{a, b\}$ of vertices $a, b \in V(G)$.

Definitions

Definition 2.2. A graph G consists of

(i) a finite non–empty set $V(G)$ of vertices and

(ii) a set $E(G)$ of edges

such that every edge $e \in E(G)$ is an unordered pair $\{a, b\}$ of vertices $a, b \in V(G)$.

We shall restrict attention to graphs with finite edge and vertex sets in this course.

Definitions

Definition 2.2. A graph G consists of

(i) a finite non–empty set $V(G)$ of vertices and

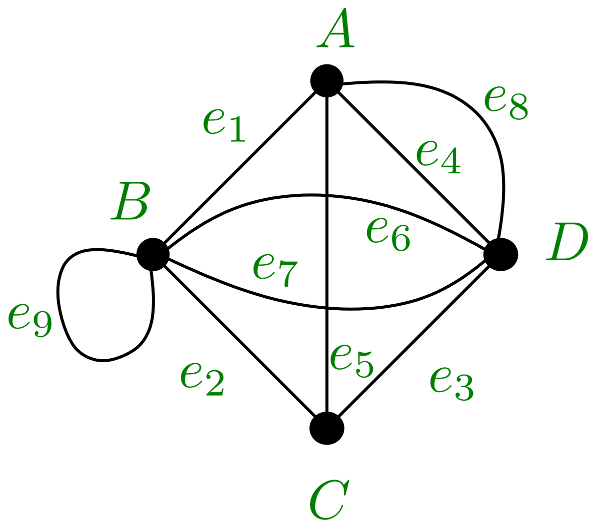
(ii) a set $E(G)$ of edges

such that every edge $e \in E(G)$ is an unordered pair $\{a, b\}$ of vertices $a, b \in V(G)$.

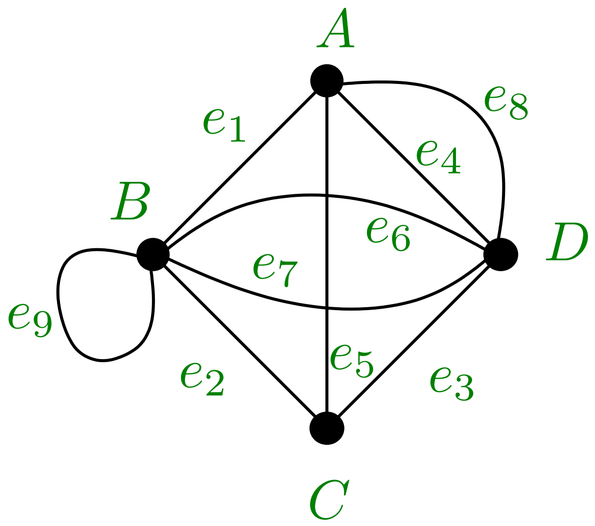
We shall restrict attention to graphs with finite edge and vertex sets in this course.

Throughout the remainder of the next two sections $G = (V, E)$ will denote a graph with (finite) vertex and edge sets V and E.

Example 2.3.



Example 2.3.



A graph must have at least one vertex but need not have any edges.

Definition 2.4. Let $G = (V, E)$ be a graph.

Definition 2.4. Let $G = (V, E)$ be a graph.

(i) Vertices a and b are adjacent if there exists an edge $e \in E$ with $e = \{a, b\}$.

Definition 2.4. Let $G = (V, E)$ be a graph.

(i) Vertices a and b are adjacent if there exists an edge $e \in E$ with $e = \{a, b\}$.

(ii) Edges e and f are adjacent if there exists a vertex $v \in V$ with $e = \{v, a\}$ and $f = \{v, b\}$, for some $a, b \in V$.

Definition 2.4. Let $G = (V, E)$ be a graph.

(i) Vertices a and b are adjacent if there exists an edge $e \in E$ with $e = \{a, b\}$.

(ii) Edges e and f are adjacent if there exists a vertex $v \in V$ with $e = \{v, a\}$ and $f = \{v, b\}$, for some $a, b \in V$.

(iii) If $e \in E$ and $e = \{c, d\}$ then e is said to be incident to c and to d and to join c and d.

Definition 2.4. Let $G = (V, E)$ be a graph.

(i) Vertices a and b are adjacent if there exists an edge $e \in E$ with $e = \{a, b\}$.

(ii) Edges e and f are adjacent if there exists a vertex $v \in V$ with $e = \{v, a\}$ and $f = \{v, b\}$, for some $a, b \in V$.

(iii) If $e \in E$ and $e = \{c, d\}$ then e is said to be incident to c and to d and to join c and d.

(iv) If a and b are vertices joined by edges $e_1, \ldots e_k$, where $k > 1$, then $e_1, \ldots e_k$ are called multiple edges.

Definition 2.4. Let $G = (V, E)$ be a graph.

(i) Vertices a and b are adjacent if there exists an edge $e \in E$ with $e = \{a, b\}$.

(ii) Edges e and f are adjacent if there exists a vertex $v \in V$ with $e = \{v, a\}$ and $f = \{v, b\}$, for some $a, b \in V$.

(iii) If $e \in E$ and $e = \{c, d\}$ then e is said to be incident to c and to d and to join c and d.

(iv) If a and b are vertices joined by edges $e_1, \ldots e_k$, where $k > 1$, then $e_1, \ldots e_k$ are called multiple edges.

(v) An edge of the form $\{a, a\}$ is called a loop.

Definition 2.4. Let $G = (V, E)$ be a graph.

(i) Vertices a and b are adjacent if there exists an edge $e \in E$ with $e = \{a, b\}$.

(ii) Edges e and f are adjacent if there exists a vertex $v \in V$ with $e = \{v, a\}$ and $f = \{v, b\}$, for some $a, b \in V$.

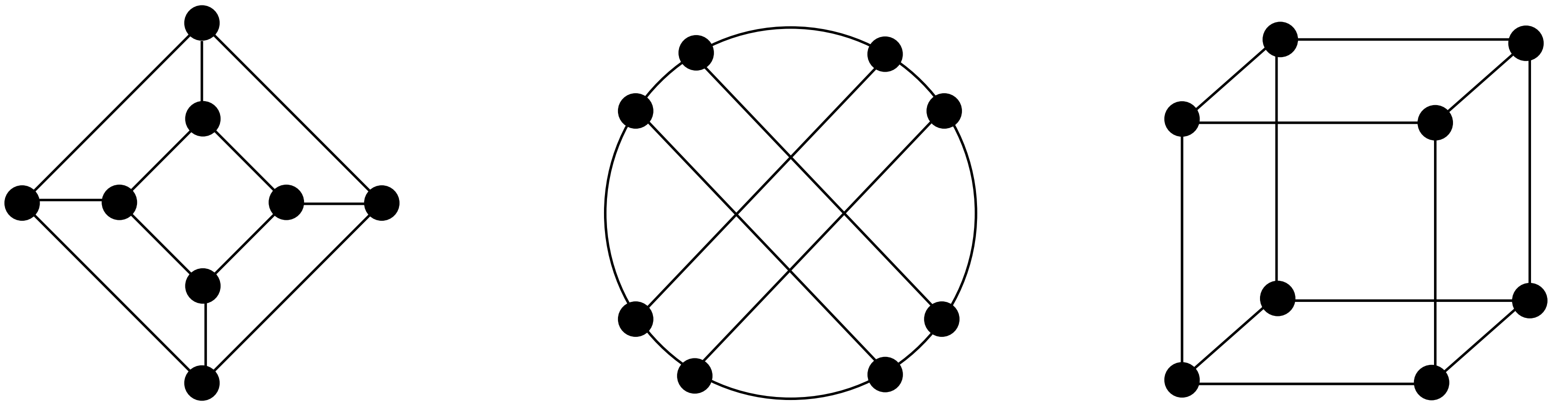
(iii) If $e \in E$ and $e = \{c, d\}$ then e is said to be incident to c and to d and to join c and d.

(iv) If a and b are vertices joined by edges $e_1, \ldots e_k$, where $k > 1$, then $e_1, \ldots e_k$ are called multiple edges.

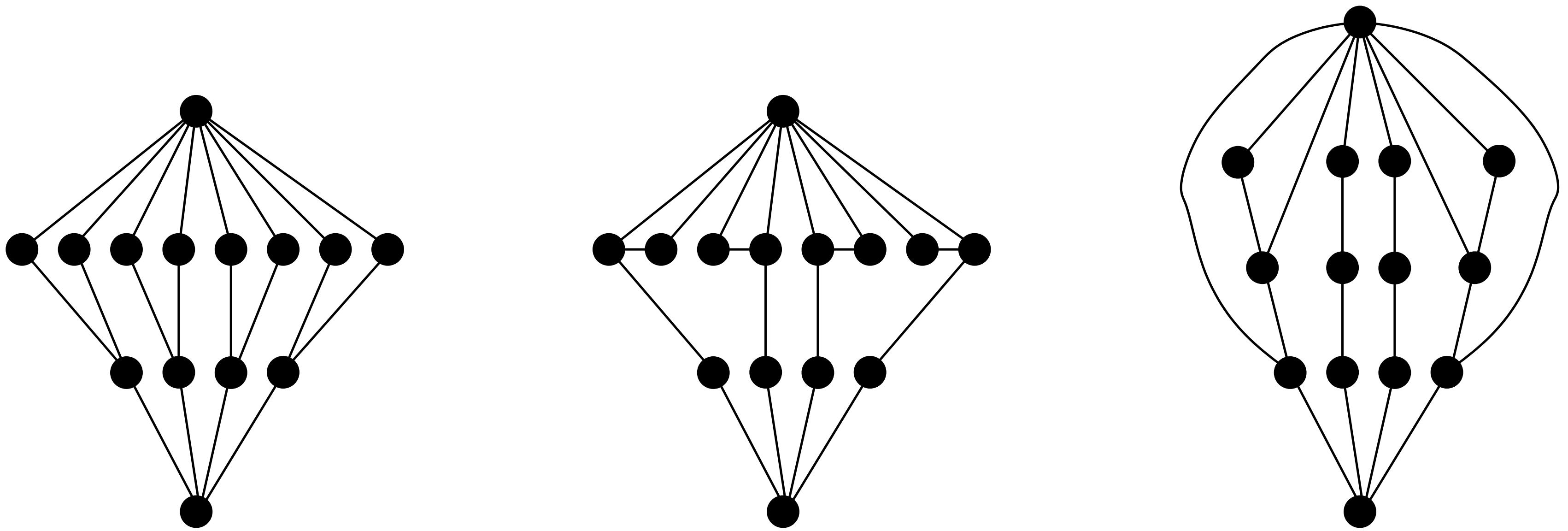
(v) An edge of the form $\{a, a\}$ is called a loop.

(vi) A graph which has no multiple edges and no loops is called a simple graph.

Example 2.5. Are these three graphs the same?



Example 2.6. What about these?



Isomorphism

Definition 2.7. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic

Isomorphism

Definition 2.7. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection

$$\phi : V_1 \longrightarrow V_2$$

Isomorphism

Definition 2.7. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection

$$\phi : V_1 \longrightarrow V_2$$

such that the number of edges joining u to v in G_1 is the same as the number of edges joining $\phi(u)$ to $\phi(v)$ in G_2, for all $u, v \in V_1$.

Isomorphism

Definition 2.7. Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there exists a bijection

$$\phi : V_1 \longrightarrow V_2$$

such that the number of edges joining u to v in G_1 is the same as the number of edges joining $\phi(u)$ to $\phi(v)$ in G_2, for all $u, v \in V_1$.

ϕ is called an isomorphism from G_1 to G_2 and we write $G_1 \cong G_2$.

Definition 2.8. The degree of a vertex u is the number of edges incident to u and is denoted $\deg(u)$ or $\text{degree}(u)$.

Definition 2.8. The degree of a vertex u is the number of edges incident to u and is denoted $\deg(u)$ or $\text{degree}(u)$.

Definition 2.9. Let G be a graph with n vertices. Order the vertices $v_1, \ldots, v_n$ so that $\deg(v_i) \leq \deg(v_{i+1})$. Then G has degree sequence

$$\langle \deg(v_1), \ldots, \deg(v_n) \rangle.$$

Definition 2.8. The degree of a vertex u is the number of edges incident to u and is denoted $\deg(u)$ or $\text{degree}(u)$.

Definition 2.9. Let G be a graph with n vertices. Order the vertices $v_1, \ldots, v_n$ so that $\deg(v_i) \leq \deg(v_{i+1})$. Then G has degree sequence

$$\langle \deg(v_1), \ldots, \deg(v_n) \rangle.$$

Definition 2.10. A graph is regular if every vertex has degree d, for some fixed $d \in \mathbb{Z}$.

Definition 2.8. The degree of a vertex u is the number of edges incident to u and is denoted $\deg(u)$ or $\text{degree}(u)$.

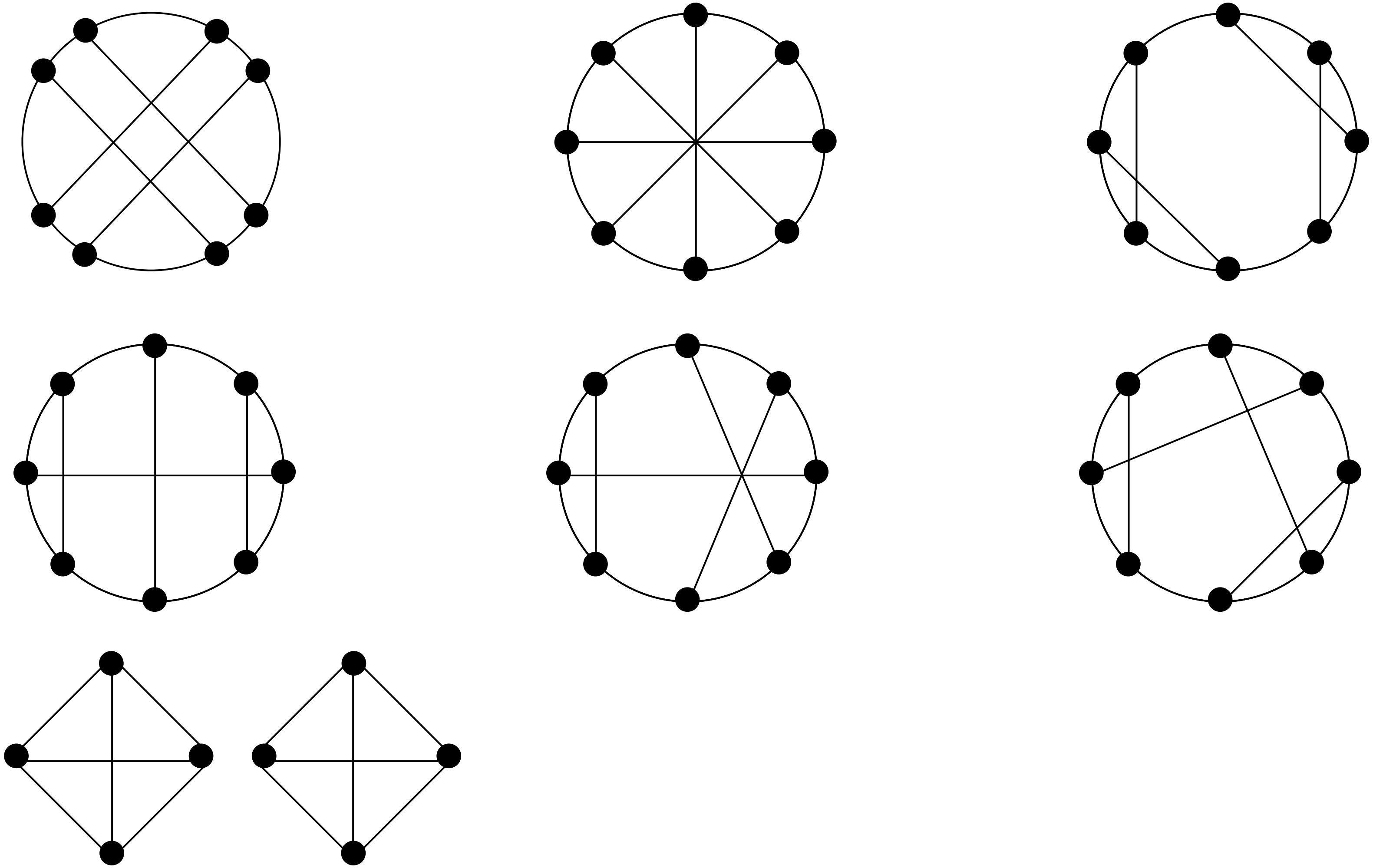
Definition 2.9. Let G be a graph with n vertices. Order the vertices $v_1, \ldots, v_n$ so that $\deg(v_i) \leq \deg(v_{i+1})$. Then G has degree sequence

$$\langle \deg(v_1), \ldots, \deg(v_n) \rangle.$$

Definition 2.10. A graph is regular if every vertex has degree d, for some fixed $d \in \mathbb{Z}$.

In this case we say the graph is regular of degree d.

Example 2.11. Graphs which are simple, have 8 vertices, 12 edges and are regular of degree 3. Are any two of these isomorphic? What are their degree sequences?



Counting degrees

G is a graph with vertices V and edges E, that is $G = (V, E)$.

Counting degrees

G is a graph with vertices V and edges E, that is $G = (V, E)$.

Lemma 2.12. *[The Handshaking Lemma]*

$$\sum_{v \in V} \deg(v) = 2|E|.$$

Counting degrees

G is a graph with vertices V and edges E, that is $G = (V, E)$.

Lemma 2.12. *[The Handshaking Lemma]*

$$\sum_{v \in V} \deg(v) = 2|E|.$$

Lemma 2.13. *Suppose that G has q vertices of odd degree. Then q is even.*

Counting degrees

G is a graph with vertices V and edges E, that is $G = (V, E)$.

Lemma 2.12. *[The Handshaking Lemma]*

$$\sum_{v \in V} \deg(v) = 2|E|.$$

Lemma 2.13. *Suppose that G has q vertices of odd degree. Then q is even.*

Corollary 2.14. *If G has n vertices and is regular of degree d then G has $nd/2$ edges.*

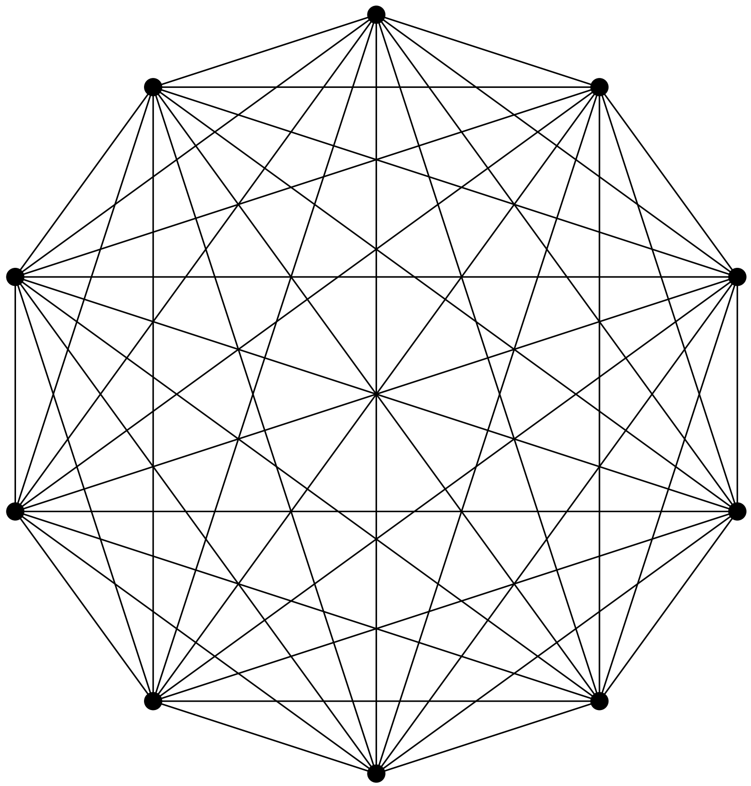
Examples of graphs

Example 2.15. The Null graph N_d, for $d \geq 1$.

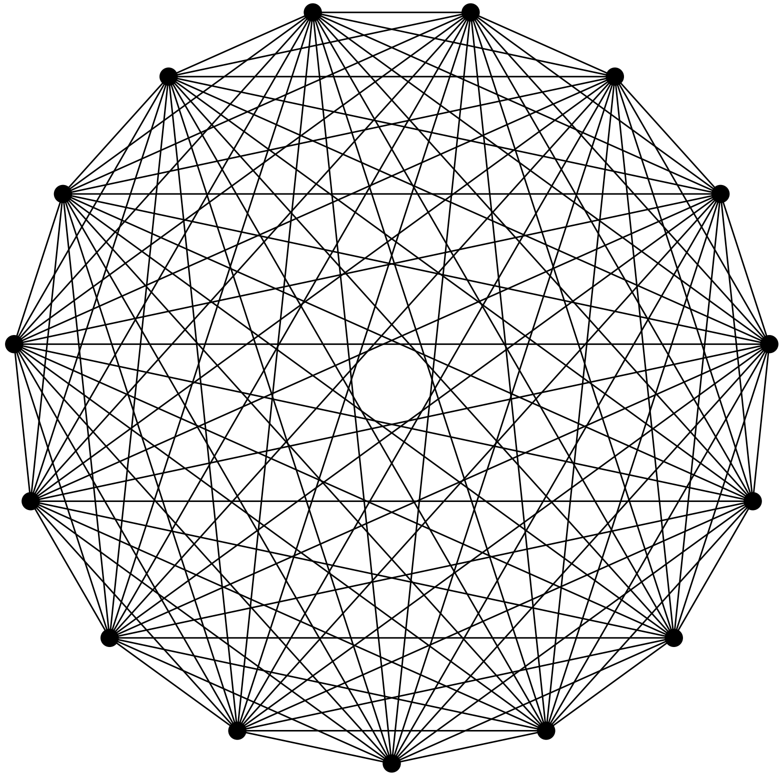
Examples of graphs

Example 2.15. The Null graph N_d, for $d \geq 1$.

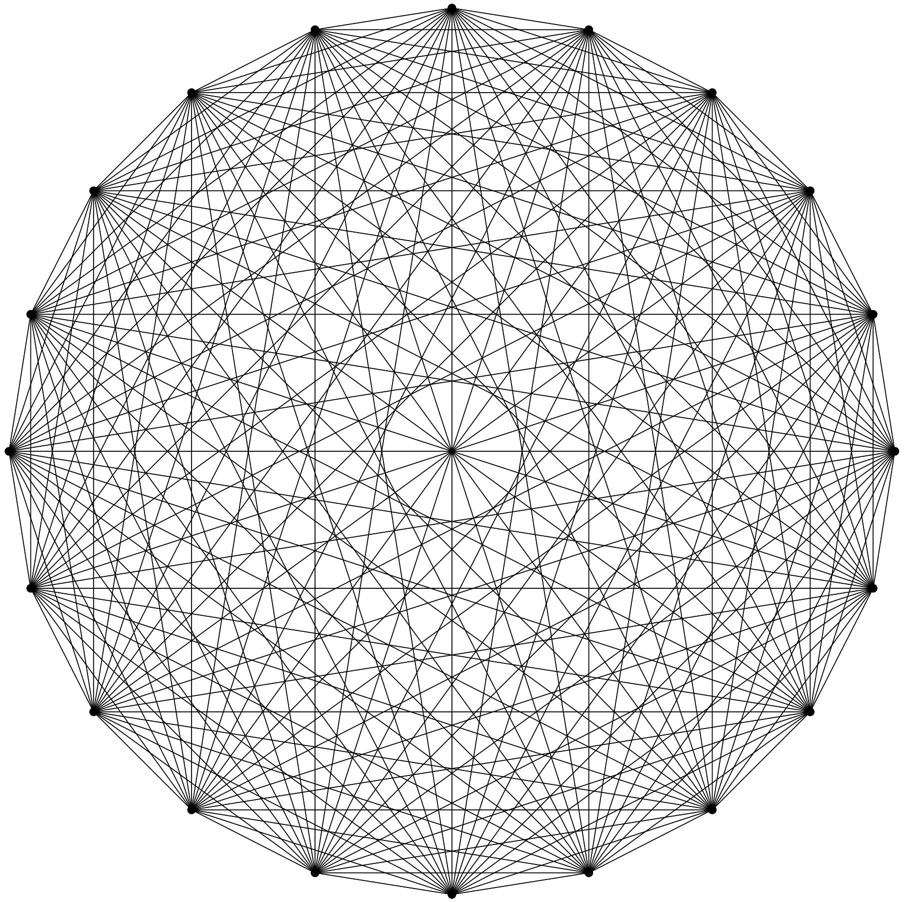
Example 2.16. The Complete graph K_d, for $d \geq 1$.



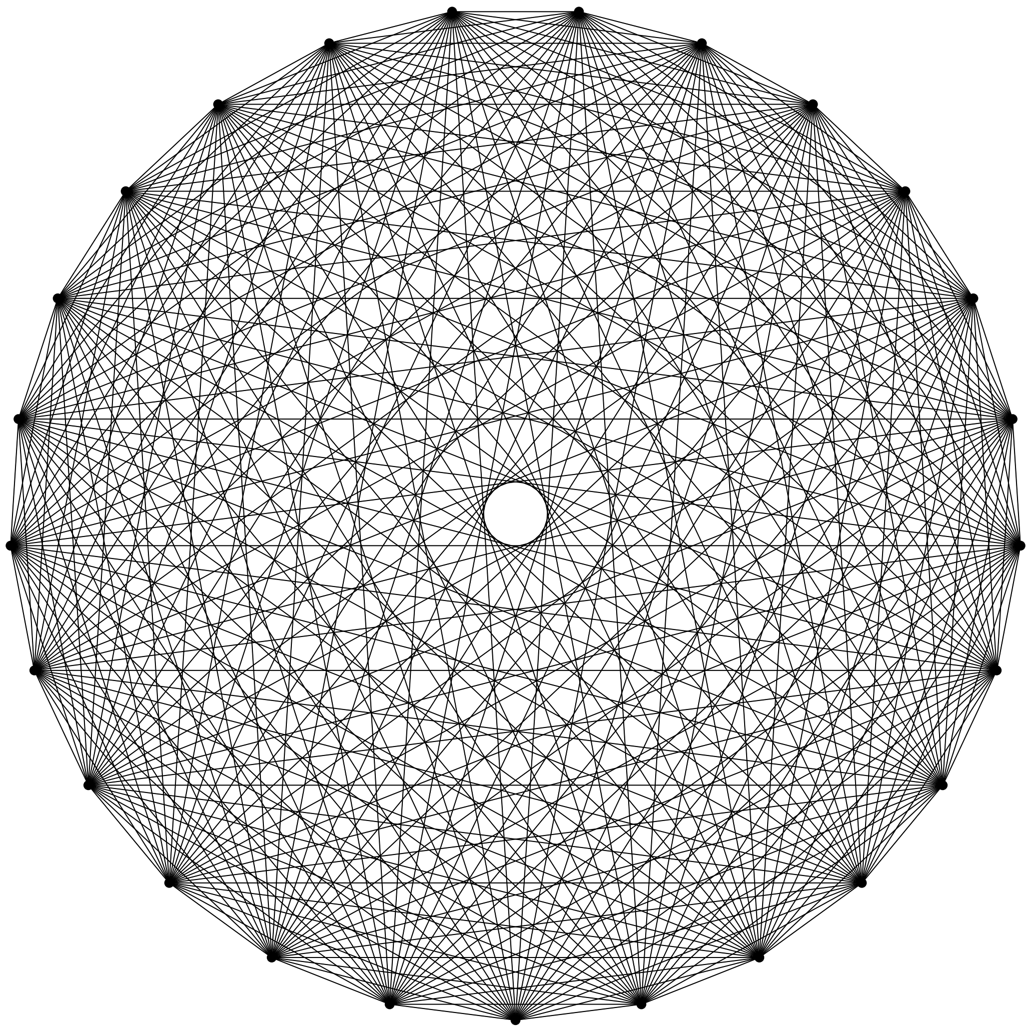
K_{10}



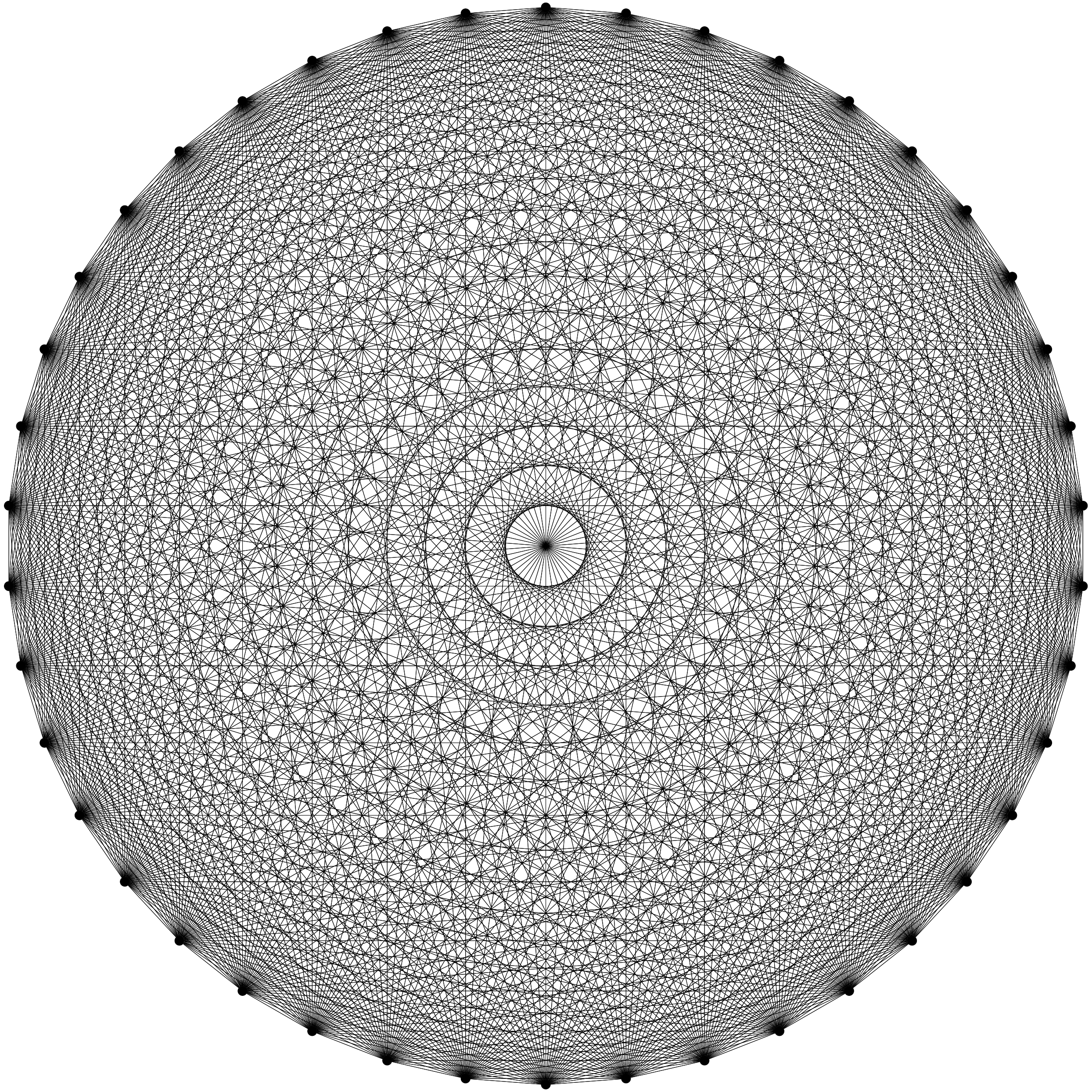
K_{15}



K_{20}



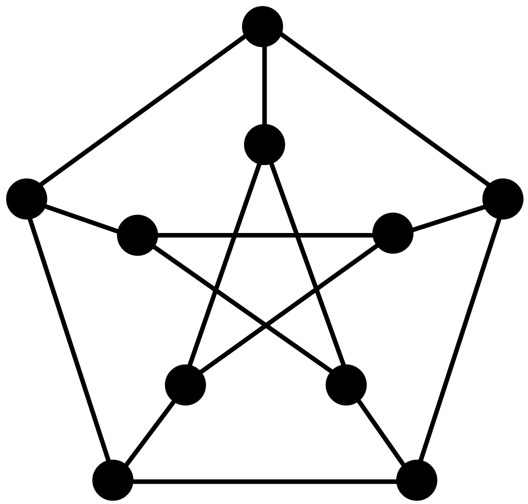
K_{25}



K_{42}

Lemma 2.17. *The complete graph* K_d *is regular of degree* $d - 1$ *and has* $d(d-1)/2$ *edges.*

Example 2.18. The Petersen graph.



Subgraphs

Definition 2.19. A subgraph of a graph $G = (V, E)$ is a graph $H = (V', E')$ such that $V' \subset V$ and $E' \subset E$.

Subgraphs

Definition 2.19. A subgraph of a graph $G = (V, E)$ is a graph $H = (V', E')$ such that $V' \subset V$ and $E' \subset E$.

Example 2.20.

1. For $d \geq 1$ we define the cycle graph C_d to be the graph with d vertices $v_1, \ldots, v_d$ and d edges $\{v_1, v_2\}, \ldots, \{v_{d-1}, v_d\}, \{v_d, v_1\}$.

Subgraphs

Definition 2.19. A subgraph of a graph $G = (V, E)$ is a graph $H = (V', E')$ such that $V' \subset V$ and $E' \subset E$.

Example 2.20.

1. For $d \geq 1$ we define the cycle graph C_d to be the graph with d vertices $v_1, \ldots, v_d$ and d edges $\{v_1, v_2\}, \ldots, \{v_{d-1}, v_d\}, \{v_d, v_1\}$.

(C_1 has one vertex v_1 and one edge $\{v_1, v_1\}$.)

Subgraphs

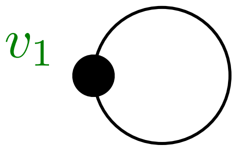
Definition 2.19. A subgraph of a graph $G = (V, E)$ is a graph $H = (V', E')$ such that $V' \subset V$ and $E' \subset E$.

Example 2.20.

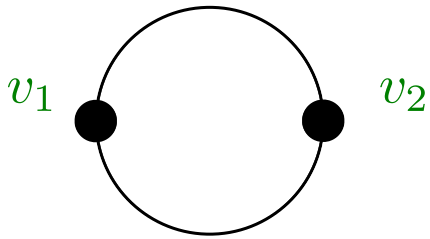
1. For $d \geq 1$ we define the cycle graph C_d to be the graph with d vertices $v_1, \ldots, v_d$ and d edges $\{v_1, v_2\}, \ldots, \{v_{d-1}, v_d\}, \{v_d, v_1\}$.

(C_1 has one vertex v_1 and one edge $\{v_1, v_1\}$.)

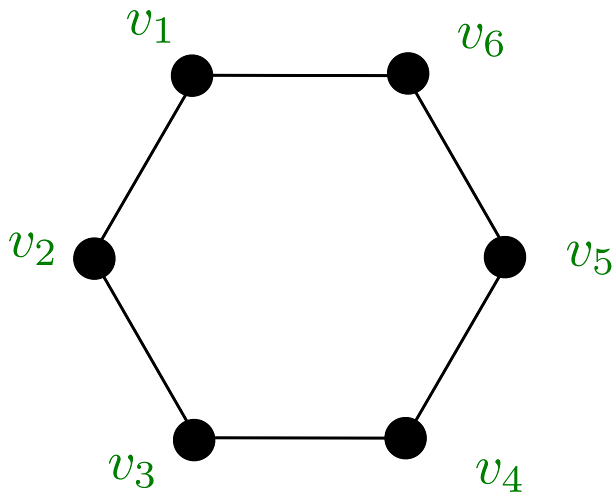
The cycle graph is regular of degree 2 and simple if $d \geq 3$.



C_1

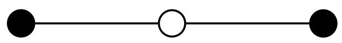


C_2

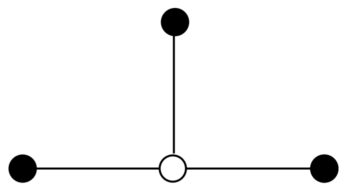


C_6

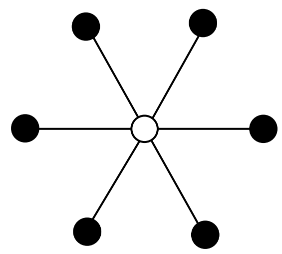
2. The **star** graphs are the graphs $K_{1,s}$, $s \geq 1$:



$K_{1,2}$



$K_{1,3}$



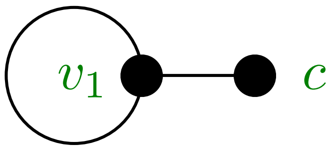
$K_{1,6}$

3. For $d \geq 1$ we define the wheel graph W_d to be the graph with $d+1$ vertices $c, v_1, \ldots, v_d$ and

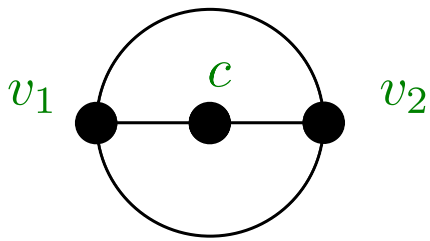
3. For $d \geq 1$ we define the wheel graph W_d to be the graph with

$d+1$ vertices $c, v_1, \ldots, v_d$ and

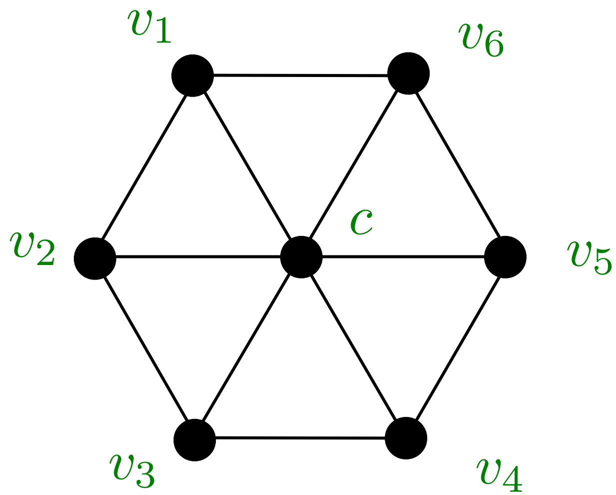
$2d$ edges $\{v_1, v_2\}, \ldots, \{v_{d-1}, v_d\}, \{v_d, v_1\}, \quad \text{and} \quad \{c, v_1\}, \ldots \{c, v_d\}$.



W_1



W_2



W_6

Walks, paths, trails, circuits and cycles

$G = (V, E)$ a graph

Definition 2.21. A sequence $v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n,$ where

Walks, paths, trails, circuits and cycles

$G = (V, E)$ a graph

Definition 2.21. A sequence $v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n,$ where

(i) $n \geq 0$ and

Walks, paths, trails, circuits and cycles

$G = (V, E)$ a graph

Definition 2.21. A sequence $v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n$, where

(i) $n \geq 0$ and

(ii) $v_i \in V$ and $e_i \in E$ and

Walks, paths, trails, circuits and cycles

$G = (V, E)$ a graph

Definition 2.21. A sequence $v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n$, where

(i) $n \geq 0$ and

(ii) $v_i \in V$ and $e_i \in E$ and

(iii) $e_i = \{v_{i-1}, v_i\}$, for $i = 1, \ldots, n$,

Walks, paths, trails, circuits and cycles

$G = (V, E)$ a graph

Definition 2.21. A sequence $v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n$, where

(i) $n \geq 0$ and

(ii) $v_i \in V$ and $e_i \in E$ and

(iii) $e_i = \{v_{i-1}, v_i\}$, for $i = 1, \ldots, n$,

is called a walk of length n.

Walks, paths, trails, circuits and cycles

$G = (V, E)$ a graph

Definition 2.21. A sequence $v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n$, where

(i) $n \geq 0$ and

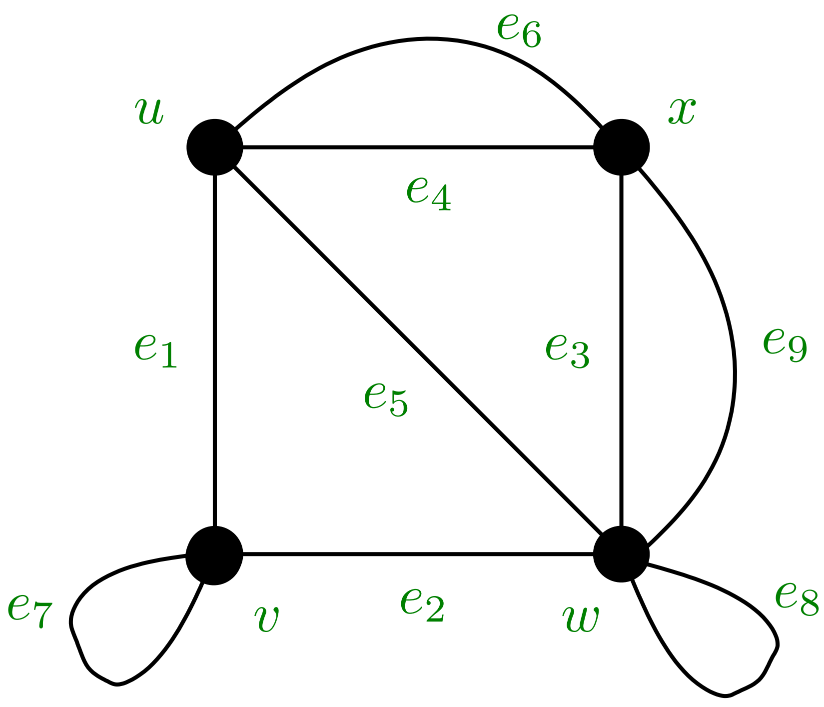
(ii) $v_i \in V$ and $e_i \in E$ and

(iii) $e_i = \{v_{i-1}, v_i\}$, for $i = 1, \ldots, n$,

is called a walk of length n.

The walk is from its initial vertex v_0 and to its terminal vertex v_n.

Example 2.22. G is the graph shown.



Definition 2.23. Let $W = v_0, e_1, v_1, \ldots, e_n, v_n$ be a walk in a graph.

(i) If $v_0 = v_n$ then W is a closed walk.

Definition 2.23. Let $W = v_0, e_1, v_1, \ldots, e_n, v_n$ be a walk in a graph.

(i) If $v_0 = v_n$ then W is a closed walk.

(ii) If $v_i \neq v_j$ when $i \neq j$, with the possible exception of $v_0 = v_n$, then W is called a path. If $v_0 \neq v_n$ the path is said to be open and if $v_0 = v_n$ it is closed.

Walks in simple graphs

In a simple graph we may write only the sequence of vertices, which we call the vertex sequence of a walk.

Walks in simple graphs

In a simple graph we may write only the sequence of vertices, which we call the vertex sequence of a walk.

For example the sequence

$$v_1, c, v_5, v_4, c, v_2$$

is the vertex sequence of a unique walk in the wheel graph W_6 shown above.

Connectedness

Definition 2.24. A graph is connected if, for any two vertices a and b there is a path from a to b.

Connectedness

Definition 2.24. A graph is connected if, for any two vertices a and b there is a path from a to b.

A graph which is not connected is called disconnected.

Connectedness

Definition 2.24. A graph is connected if, for any two vertices a and b there is a path from a to b.

A graph which is not connected is called disconnected.

Lemma 2.25. *Let a and b be vertices of a graph. There is an path from a to b if and only if there is a walk from a to b.*

Connected Component

Definition 2.26. A connected component of a graph G is a subgraph H of G such that

Connected Component

Definition 2.26. A connected component of a graph G is a subgraph H of G such that

1. H is a connected subgraph of G and

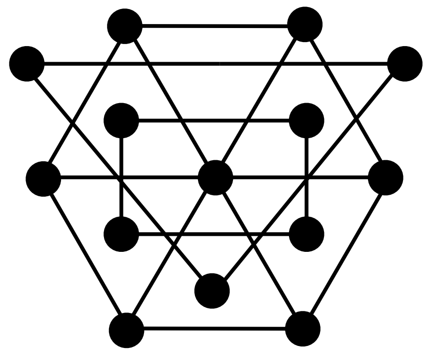
Connected Component

Definition 2.26. A connected component of a graph G is a subgraph H of G such that

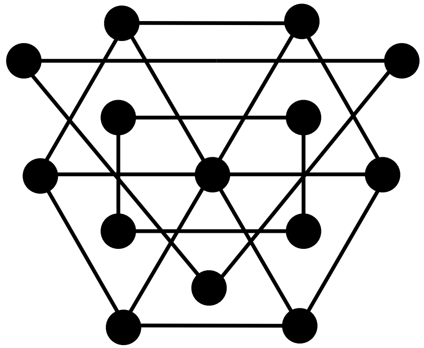
1. H is a connected subgraph of G and

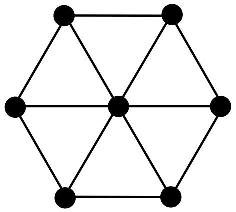
2. H is not contained in any larger connected subgraph of G.

The graph G below has 3 connected components A, B and C, as shown.

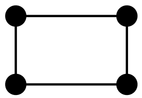


The graph G below has 3 connected components A, B and C, as shown.

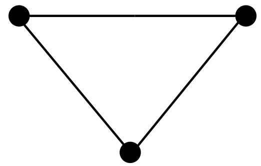




A



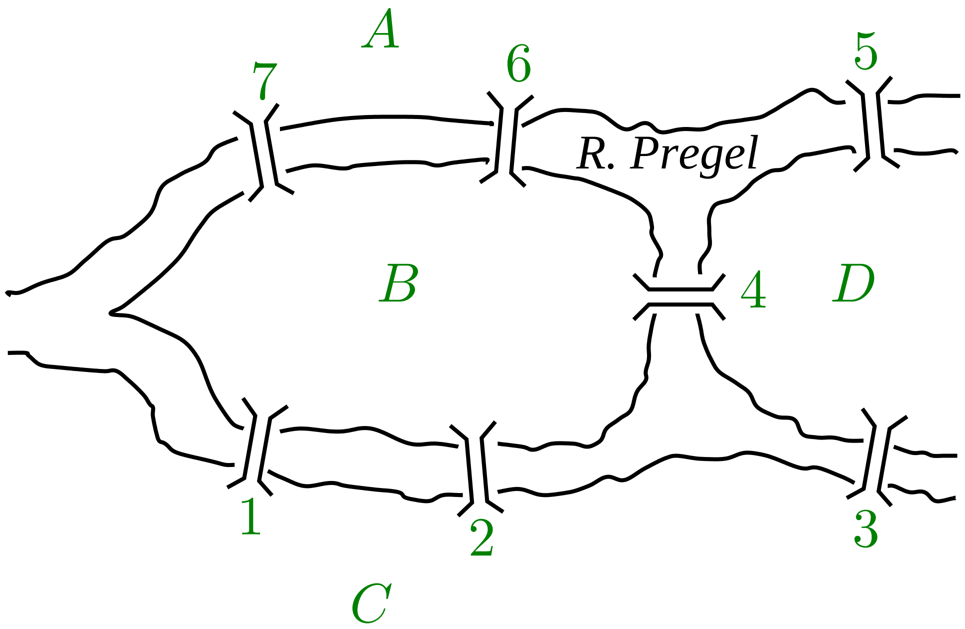
B



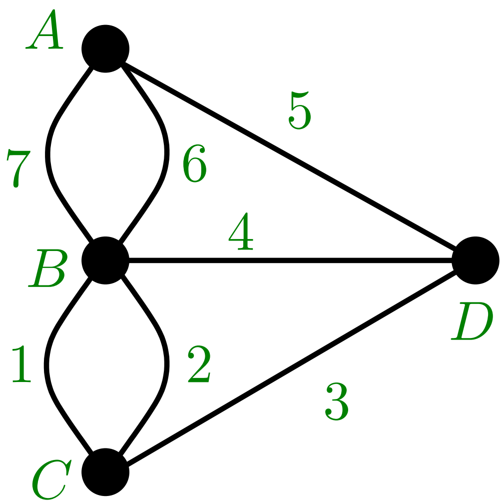
C

Eulerian graphs

The Königsberg bridge problem.



The River Pregel in Königsberg



A graph of the Königsberg bridges

Definition 2.27. A walk in which no edges are repeated is called a trail. A closed walk which is a trail is called a circuit.

Definition 2.27. A walk in which no edges are repeated is called a trail. A closed walk which is a trail is called a circuit.

Definition 2.28. 1. A trail containing every edge of a graph is called an Eulerian trail.

Definition 2.27. A walk in which no edges are repeated is called a trail. A closed walk which is a trail is called a circuit.

Definition 2.28. 1. A trail containing every edge of a graph is called an Eulerian trail.

2. A circuit containing every edge of a graph is called an Eulerian circuit.

Definition 2.27. A walk in which no edges are repeated is called a trail. A closed walk which is a trail is called a circuit.

Definition 2.28. 1. A trail containing every edge of a graph is called an Eulerian trail.

2. A circuit containing every edge of a graph is called an Eulerian circuit.

3. A graph is called semi–Eulerian if it is connected and has an Eulerian trail.

Definition 2.27. A walk in which no edges are repeated is called a trail. A closed walk which is a trail is called a circuit.

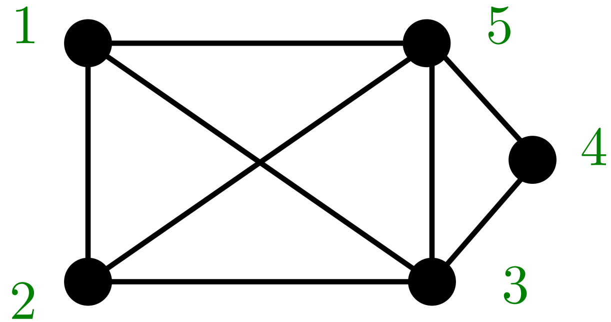
Definition 2.28. 1. A trail containing every edge of a graph is called an Eulerian trail.

2. A circuit containing every edge of a graph is called an Eulerian circuit.

3. A graph is called semi–Eulerian if it is connected and has an Eulerian trail.

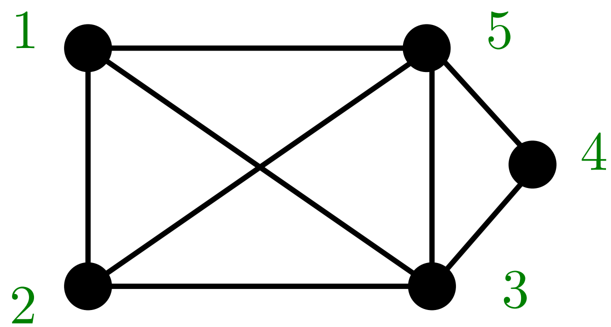
4. A graph is called Eulerian if it is connected and has an Eulerian circuit.

Example 2.29. 1. The walk $1, 2, 3, 1, 5, 4, 3, 5, 2$ is a semi–Eulerian trail in the graph G_1 below.



G_1

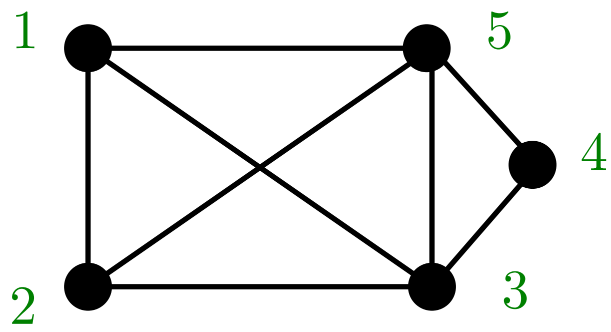
Example 2.29. 1. The walk $1, 2, 3, 1, 5, 4, 3, 5, 2$ is a semi–Eulerian trail in the graph G_1 below.



G_1

Therefore G_1 is semi–Eulerian.

Example 2.29. 1. The walk $1, 2, 3, 1, 5, 4, 3, 5, 2$ is a semi–Eulerian trail in the graph G_1 below.

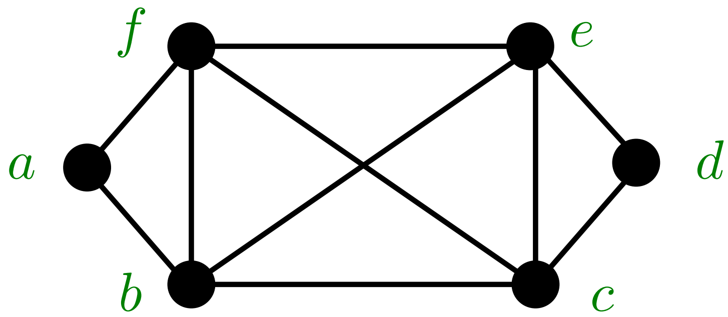


G_1

Therefore G_1 is semi–Eulerian.

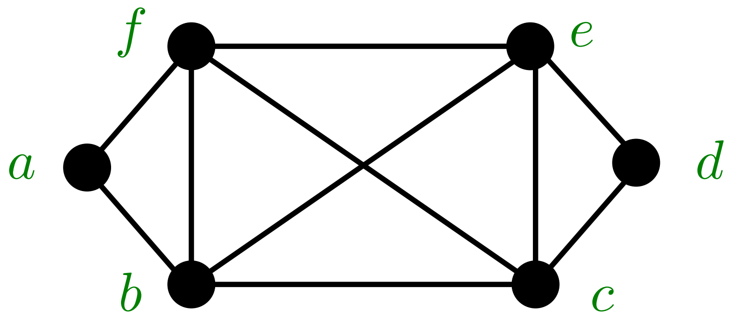
Does G_1 have an Eulerian circuit?

2. The walk $a, b, c, d, e, f, b, e, c, f, a$ is an Eulerian circuit in the graph G_2 below.



G_2

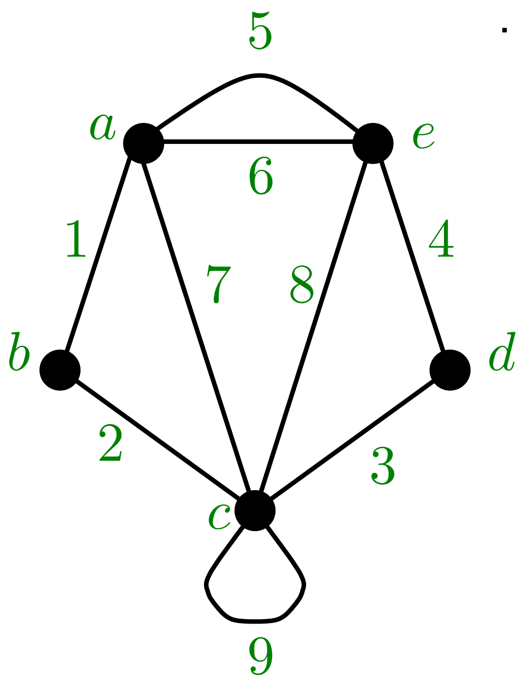
2. The walk $a, b, c, d, e, f, b, e, c, f, a$ is an Eulerian circuit in the graph G_2 below.



G_2

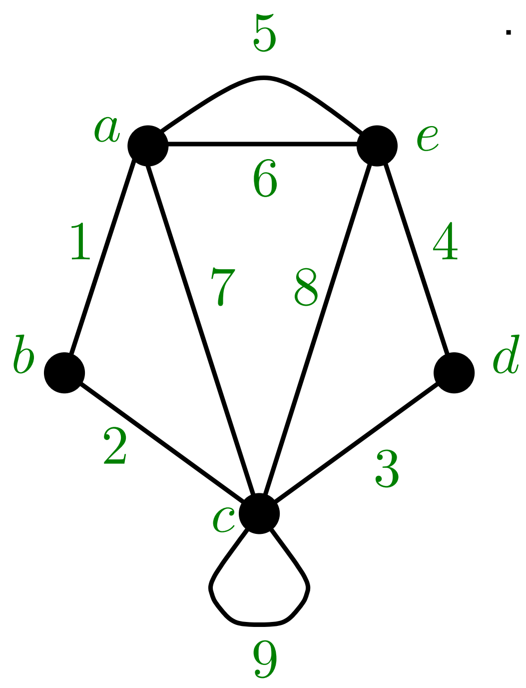
Therefore G_2 is Eulerian.

3. The walk $a, 1, b, 2, c, 3, d, 4, e, 5, a, 7, c, 9, c, 8, e, 6, a$ is an Eulerian circuit in the graph G_3 below.



G_3

3. The walk $a, 1, b, 2, c, 3, d, 4, e, 5, a, 7, c, 9, c, 8, e, 6, a$ is an Eulerian circuit in the graph G_3 below.



G_3

Therefore G_3 is Eulerian.

Theorem 2.30. *[Euler, 1736] If* G *is an Eulerian graph then every vertex of* G *has even degree.*

Theorem 2.30. *[Euler, 1736] If G is an Eulerian graph then every vertex of G has even degree.*

Example 2.31. 1. There is no Eulerian circuit for the graph of Example 2.29.1 above: this graph has verices of odd degree.

Theorem 2.30. *[Euler, 1736] If G is an Eulerian graph then every vertex of G has even degree.*

Example 2.31. 1. There is no Eulerian circuit for the graph of Example 2.29.1 above: this graph has verices of odd degree.

2. The graph of the Königsberg bridges problem has vertices of odd degree so is not Eulerian.

Theorem 2.30. *[Euler, 1736] If G is an Eulerian graph then every vertex of G has even degree.*

Example 2.31. 1. There is no Eulerian circuit for the graph of Example 2.29.1 above: this graph has verices of odd degree.

2. The graph of the Königsberg bridges problem has vertices of odd degree so is not Eulerian.

3. The graph K_d is not Eulerian if d is even.

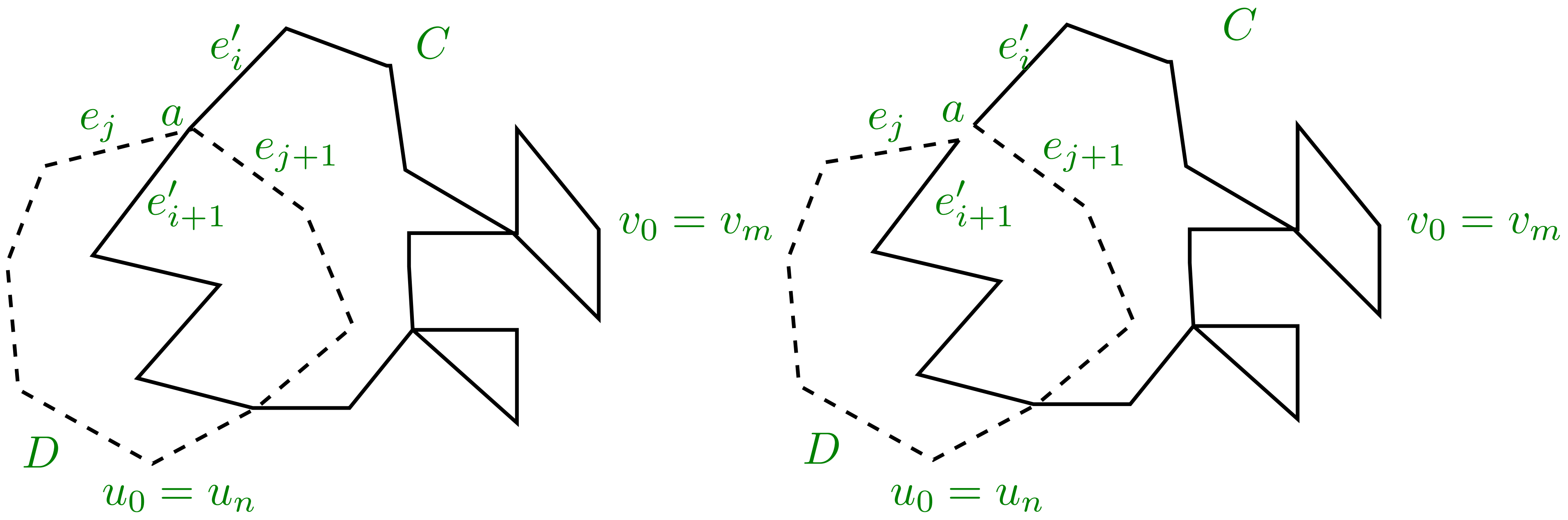
Lemma 2.32. *Let* G *be a graph such that every vertex of* G *has even degree. If* $v \in V(G)$ *with* $\deg(v) > 0$ *then* v *lies in a circuit of positive length.*

Lemma 2.32. *Let* G *be a graph such that every vertex of* G *has even degree. If* $v \in V(G)$ *with* $\deg(v) > 0$ *then* v *lies in a circuit of positive length.*

Theorem 2.33. *Let* G *be a connected graph. Then* G *is Eulerian if and only if every vertex of* G *has even degree.*

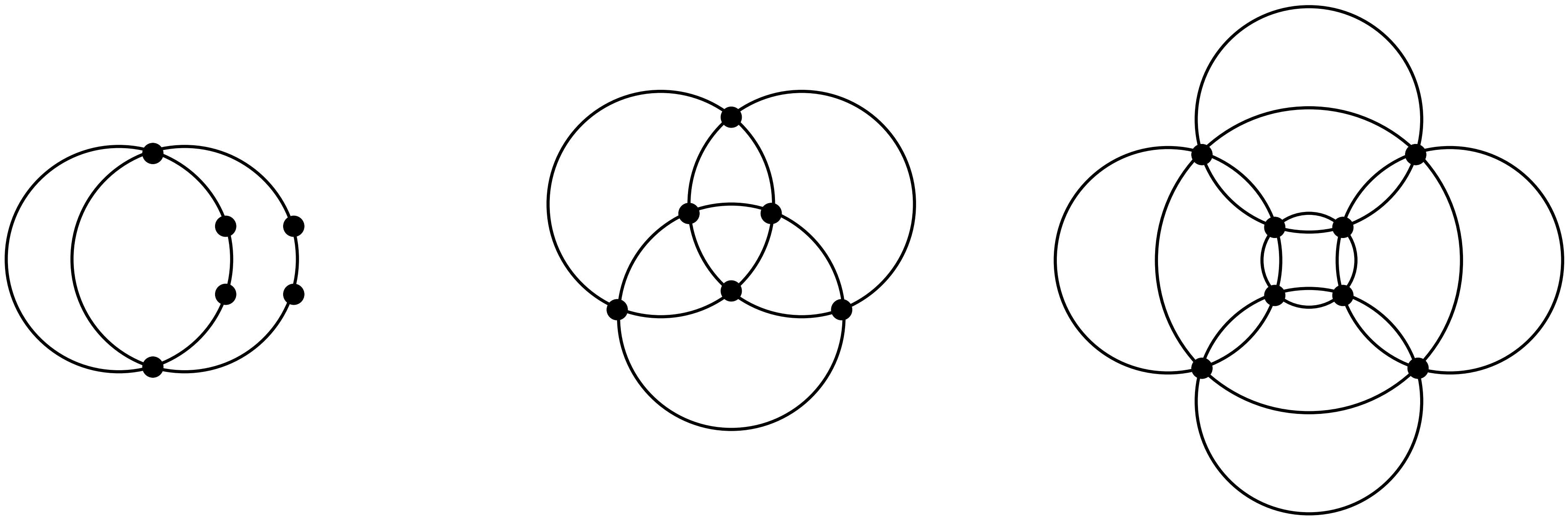
Lemma 2.32. *Let G be a graph such that every vertex of G has even degree. If $v \in V(G)$ with $\deg(v) > 0$ then v lies in a circuit of positive length.*

Theorem 2.33. *Let G be a connected graph. Then G is Eulerian if and only if every vertex of G has even degree.*



Example 2.34.

Example 2.34.



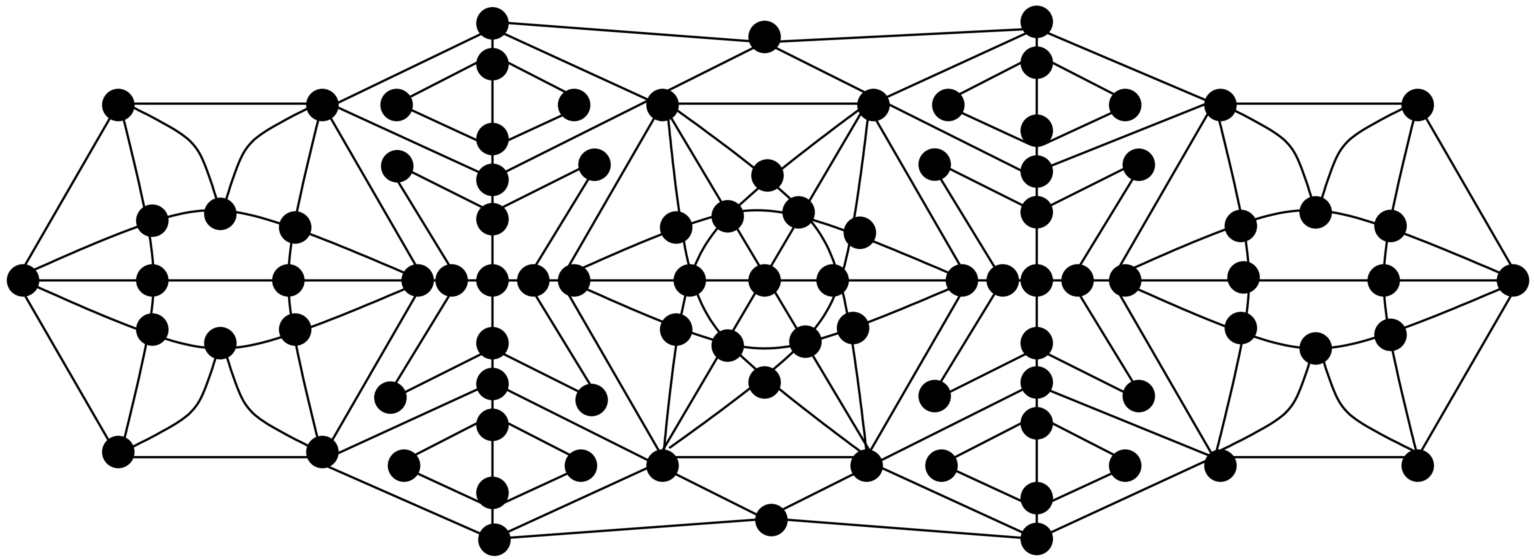
Theorem 2.35. *A connected graph is semi-Eulerian but not Eulerian if and only if precisely* 2 *of its vertices have odd degree.*

Theorem 2.35. *A connected graph is semi-Eulerian but not Eulerian if and only if precisely* 2 *of its vertices have odd degree.*

Example 2.36. The following graph has exactly 2 vertices of odd degree and is therefore semi–Eulerian.

Theorem 2.35. *A connected graph is semi-Eulerian but not Eulerian if and only if precisely 2 of its vertices have odd degree.*

Example 2.36. The following graph has exactly 2 vertices of odd degree and is therefore semi–Eulerian.



Hamiltonian Graphs

People at a party are to be seated at a circular table. Is it possible to arrange the seating so that everyone sits next to two people they know?

Hamiltonian Graphs

People at a party are to be seated at a circular table. Is it possible to arrange the seating so that everyone sits next to two people they know?

Definition 2.37.

1. A path containing every vertex of a graph is called a Hamiltonian path.

Hamiltonian Graphs

People at a party are to be seated at a circular table. Is it possible to arrange the seating so that everyone sits next to two people they know?

Definition 2.37.

1. A path containing every vertex of a graph is called a Hamiltonian path.

2. A closed path containing every vertex of a graph is called a Hamiltonian closed path.

Hamiltonian Graphs

People at a party are to be seated at a circular table. Is it possible to arrange the seating so that everyone sits next to two people they know?

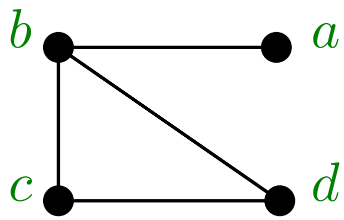
Definition 2.37.

1. A path containing every vertex of a graph is called a Hamiltonian path.
2. A closed path containing every vertex of a graph is called a Hamiltonian closed path.
3. A graph is called semi-Hamiltonian if it has a Hamiltonian path and Hamiltonian if it has a Hamiltonian closed path.

Example 2.38.

1. The walk a, b, c, d is a Hamiltonian path in the graph G_1 below.

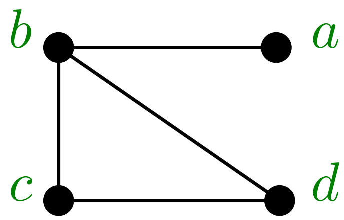
G_1:



Example 2.38.

1. The walk a, b, c, d is a Hamiltonian path in the graph G_1 below.

G_1:

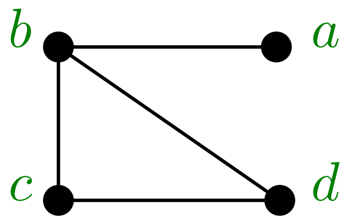


Therefore G_1 is semi-Hamiltonian.

Example 2.38.

1. The walk a, b, c, d is a Hamiltonian path in the graph G_1 below.

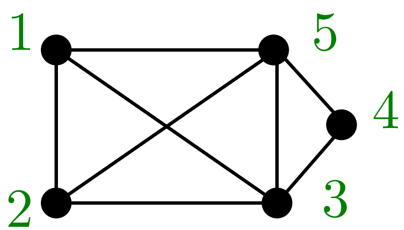
G_1:



Therefore G_1 is semi-Hamiltonian.

2. The walk $1, 2, 3, 4, 5, 1$ is a Hamiltonian closed path in the graph G_2 below.

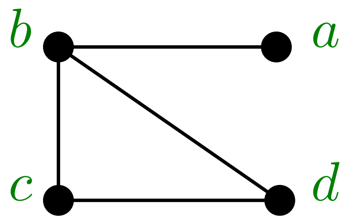
G_2:



Example 2.38.

1. The walk a, b, c, d is a Hamiltonian path in the graph G_1 below.

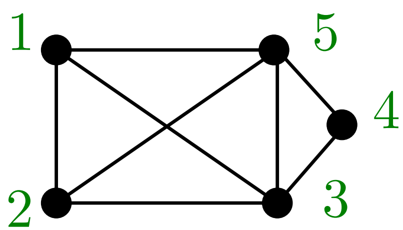
G_1:



   Therefore G_1 is semi-Hamiltonian.

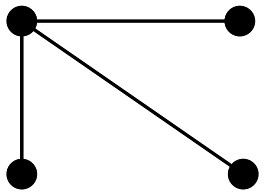
2. The walk $1, 2, 3, 4, 5, 1$ is a Hamiltonian closed path in the graph G_2 below.

G_2:



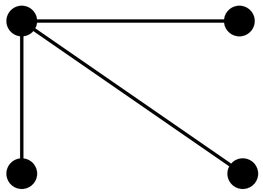
Therefore G_2 is Hamiltonian.

3. The graph G_3 below is not semi-Hamiltonian (and therefore not Hamiltonian).

G_3: 

3. The graph G_3 below is not semi-Hamiltonian (and therefore not Hamiltonian).

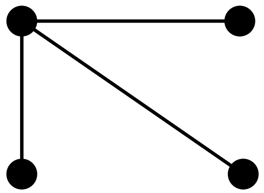
G_3:



4. The complete graph K_2 is semi-Hamiltonian but not Hamiltonian.

3. The graph G_3 below is not semi-Hamiltonian (and therefore not Hamiltonian).

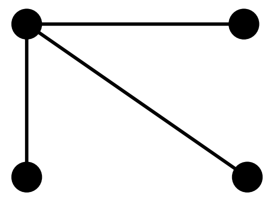
G_3:



4. The complete graph K_2 is semi-Hamiltonian but not Hamiltonian.

For $d \neq 2$ the graphs K_d are Hamiltonian.

3. The graph G_3 below is not semi-Hamiltonian (and therefore not Hamiltonian).

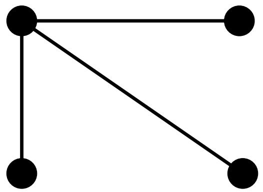
G_3: 

4. The complete graph K_2 is semi-Hamiltonian but not Hamiltonian.

For $d \neq 2$ the graphs K_d are Hamiltonian.

5. The cycle graphs are Hamiltonian for $d \geq 1$.

3. The graph G_3 below is not semi-Hamiltonian (and therefore not Hamiltonian).

G_3: 

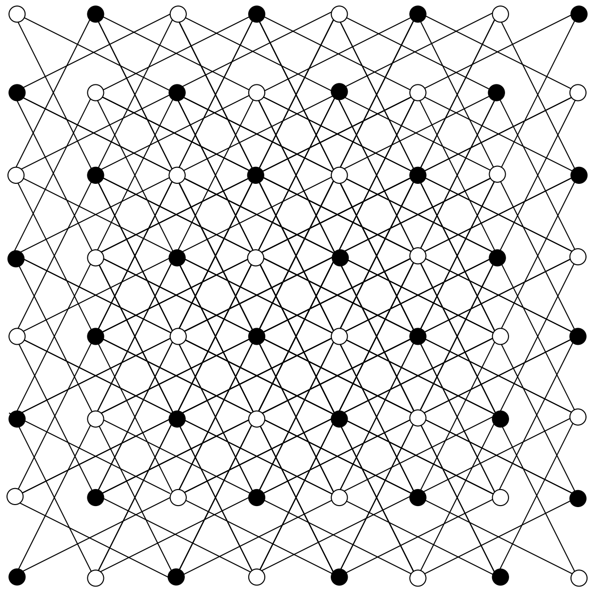
4. The complete graph K_2 is semi-Hamiltonian but not Hamiltonian.

For $d \neq 2$ the graphs K_d are Hamiltonian.

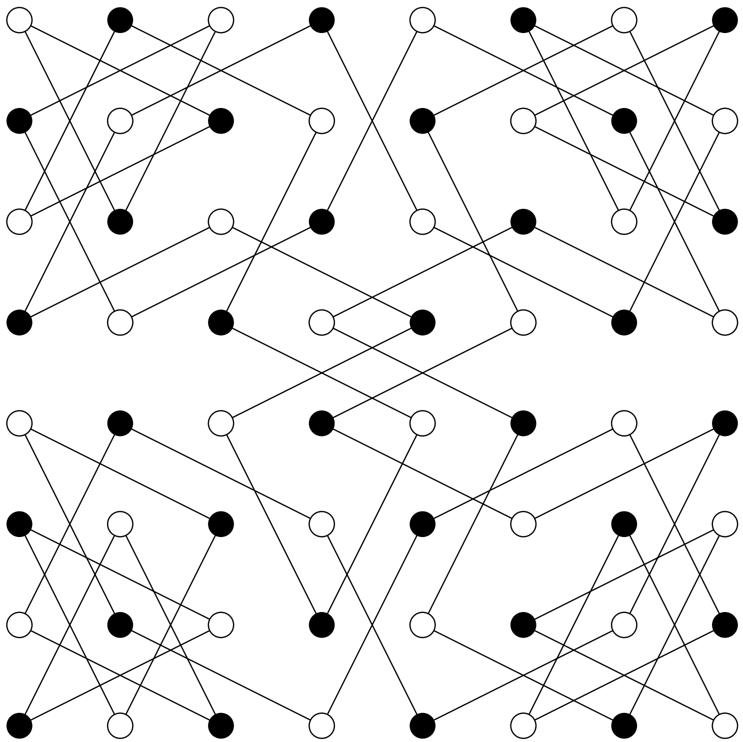
5. The cycle graphs are Hamiltonian for $d \geq 1$.

6. The wheel graph W_d is Hamiltonian for $d \geq 2$.

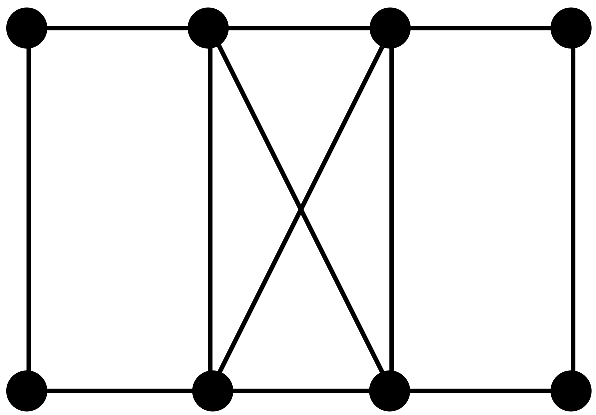
7. Construct a graph with one vertex corresponding to each square of a chessboard and an edge joining two vertices if a knight can move from one to the other. We call this the knight's move graph.



A Hamiltonian closed path for the knight's move graph

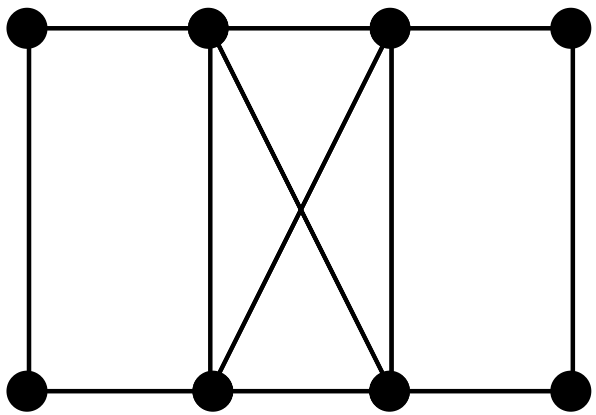


Example 2.39.

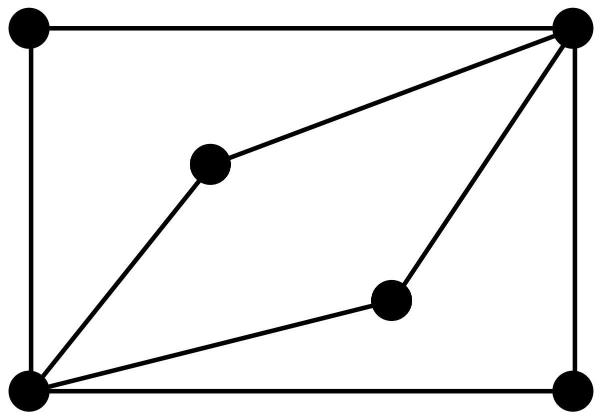


A graph which is Hamiltonian and Eulerian

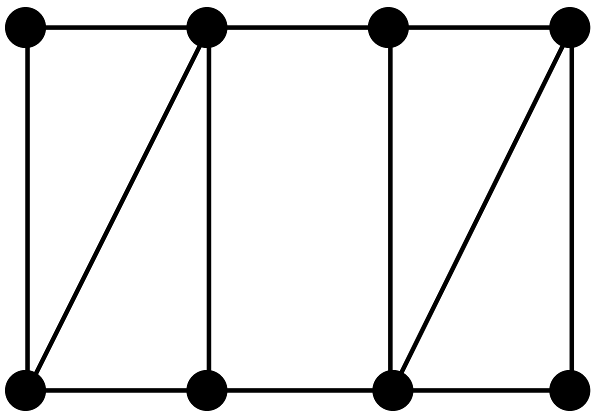
Example 2.39.



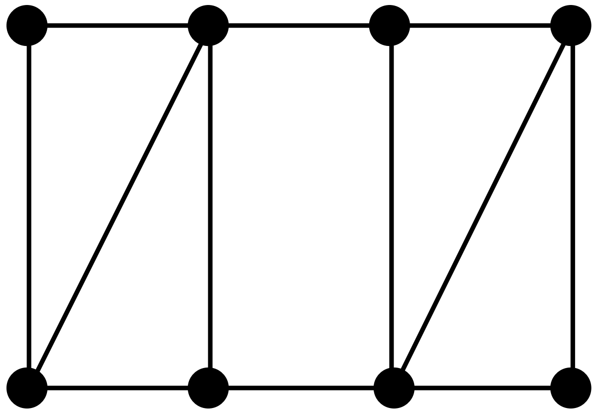
A graph which is Hamiltonian and Eulerian



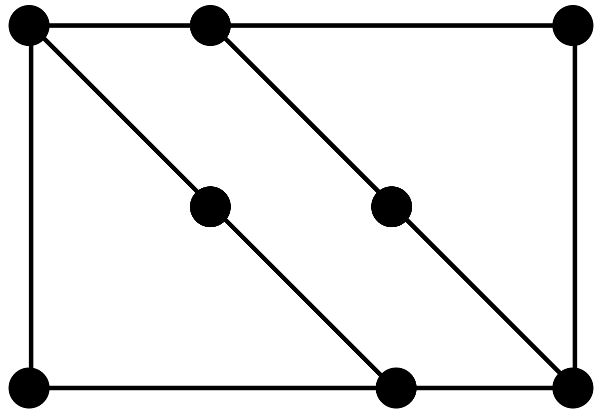
A graph which is Eulerian and non-Hamiltonian



A graph which is Hamiltonian and non-Eulerian



A graph which is Hamiltonian and non-Eulerian



A graph which is non-Eulerian and non-Hamiltonian

Trees

Definition 2.40. A closed path of length at least 1 is called a cycle.

Trees

Definition 2.40. A closed path of length at least 1 is called a cycle.

1. A forest is a graph with no cycle.

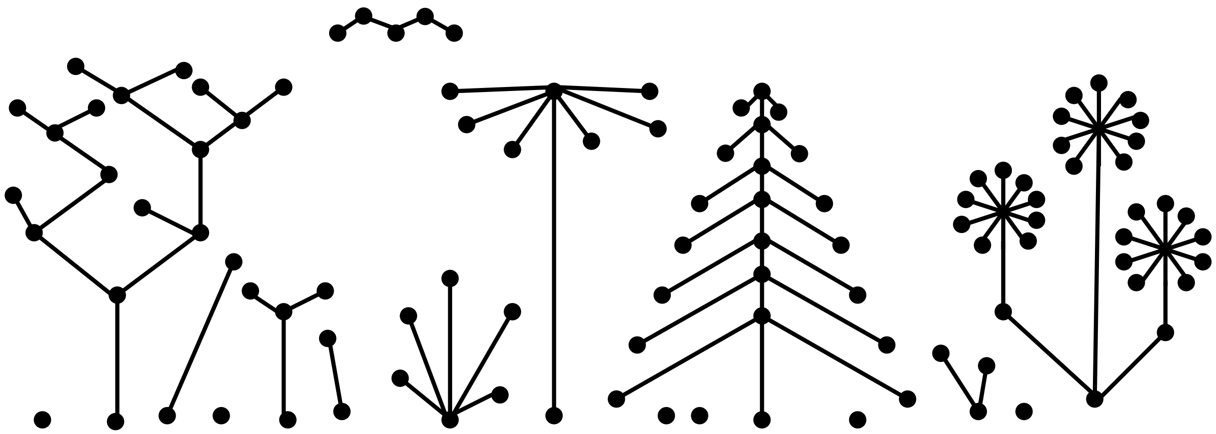
Trees

Definition 2.40. A closed path of length at least 1 is called a cycle.

1. A forest is a graph with no cycle.
2. A tree is a connected graph with no cycle.

Example 2.41.

A forest:



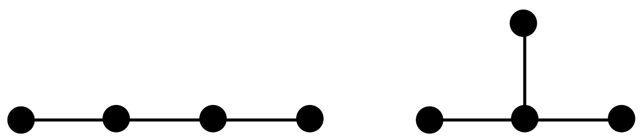
Example 2.42.

1. There is only one tree with one vertex, $N_1 = P_1$. There is only one tree with 2 vertices, $K_2 = P_2$. There is only one tree with 3 vertices, namely P_3.

Example 2.42.

1. There is only one tree with one vertex, $N_1 = P_1$. There is only one tree with 2 vertices, $K_2 = P_2$. There is only one tree with 3 vertices, namely P_3.

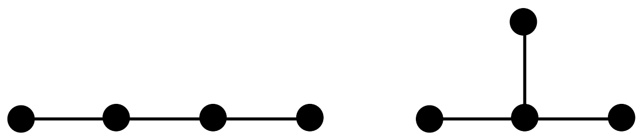
2. There are 2 trees with 4 vertices:



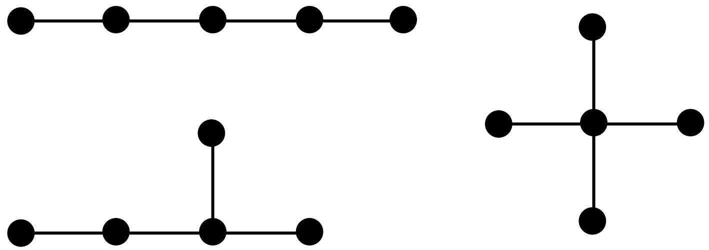
Example 2.42.

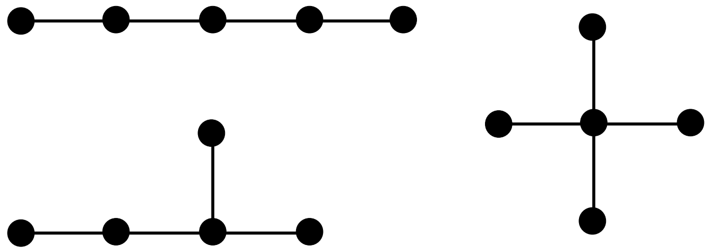
1. There is only one tree with one vertex, $N_1 = P_1$. There is only one tree with 2 vertices, $K_2 = P_2$. There is only one tree with 3 vertices, namely P_3.

2. There are 2 trees with 4 vertices:



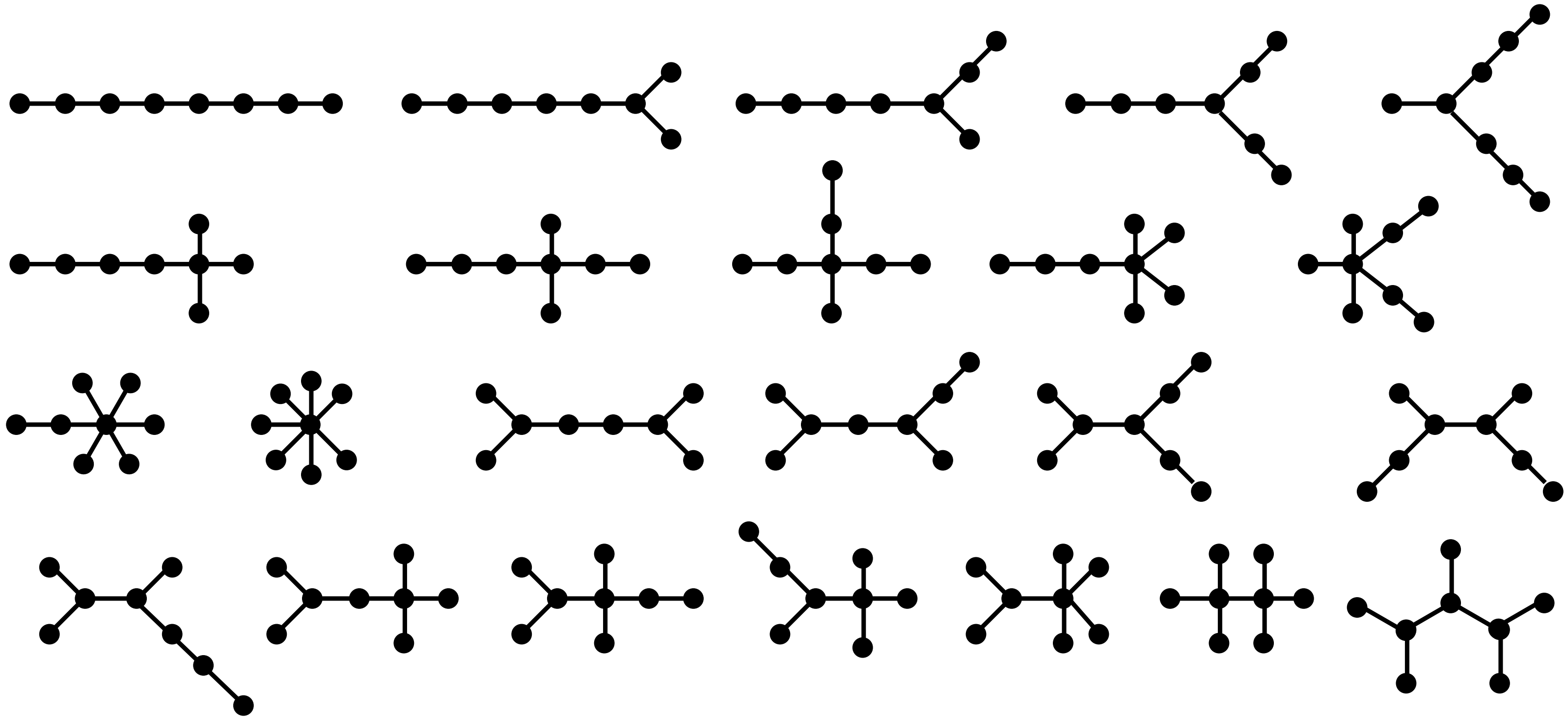
3. There are 3 trees with 5 vertices.





4. There are 6 trees with 6 vertices and 11 trees with 7 vertices (see the Exercises).

5. There are 23 trees with 8 vertices:



Characterising trees

Lemma 2.43. *If a graph* G *contains two distinct paths from vertices* u *to* v *then* G *contains a cycle.*

Characterising trees

Lemma 2.43. *If a graph* G *contains two distinct paths from vertices* u *to* v *then* G *contains a cycle.*

Theorem 2.44. *A graph* G *is a tree if and only if there is exactly one path from* u *to* v*, for all pairs* u, v *of vertices of* G*.*

Characterising trees

Lemma 2.43. *If a graph G contains two distinct paths from vertices u to v then G contains a cycle.*

Theorem 2.44. *A graph G is a tree if and only if there is exactly one path from u to v, for all pairs u, v of vertices of G.*

Theorem 2.45. *Let G be a with n vertices. Then G is a tree if and only if G is connected and has $n - 1$ edges.*

Spanning Trees

Definition 2.46.
Let G be a graph. A spanning tree for G is a subgraph of G which

Spanning Trees

Definition 2.46.
Let G be a graph. A spanning tree for G is a subgraph of G which

1. is a tree and

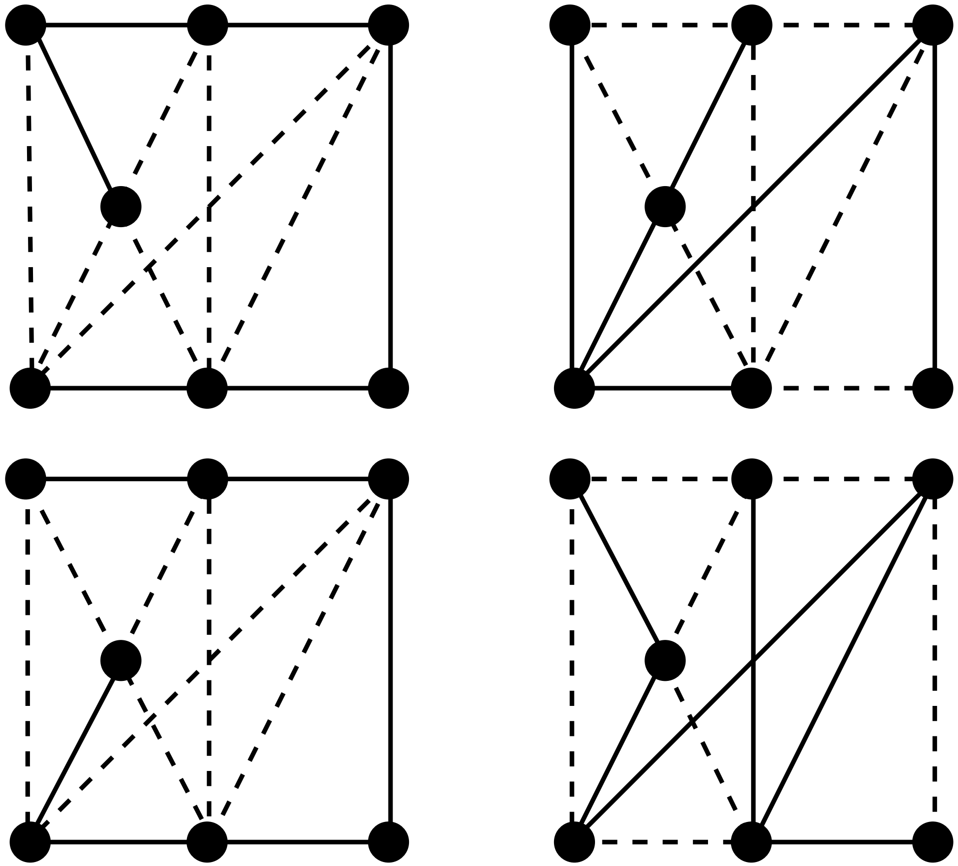
Spanning Trees

Definition 2.46.
Let G be a graph. A spanning tree for G is a subgraph of G which

1. is a tree and

2. contains every vertex of G.

Example 2.47. In the diagrams below the solid lines indicate some of the spanning trees of the graph shown: there are many more.



Theorem 2.48. *Every connected graph has a spanning tree.*

The cut-down algorithm

Given a connected graph G to construct a spanning tree:

The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

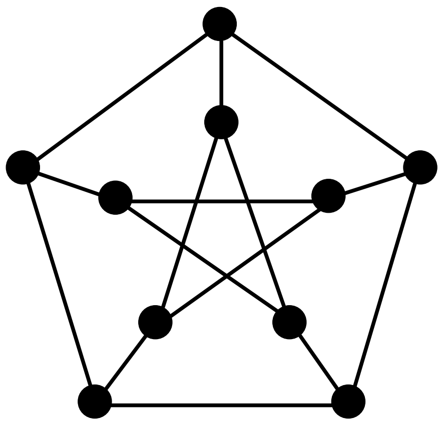
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



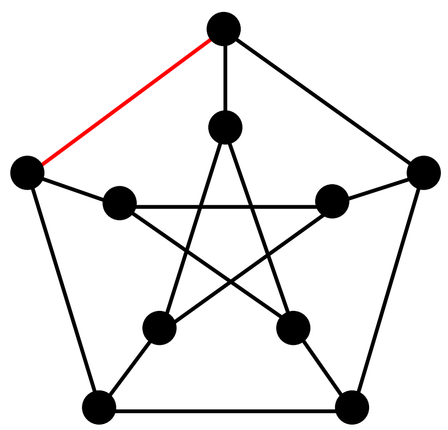
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



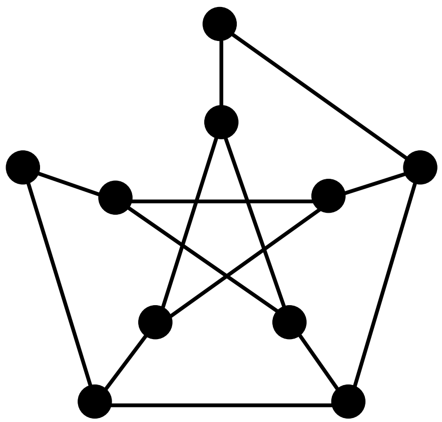
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



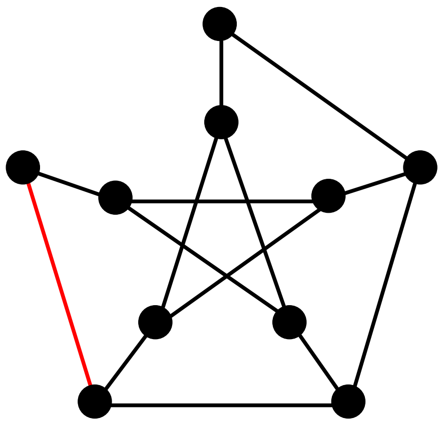
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



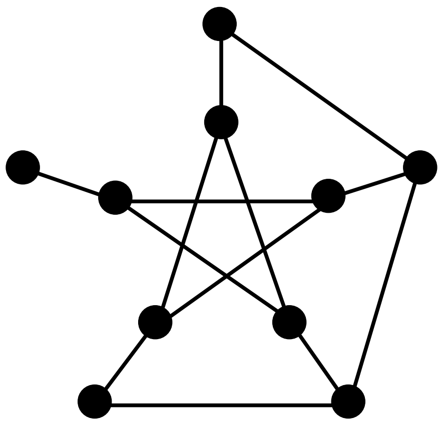
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



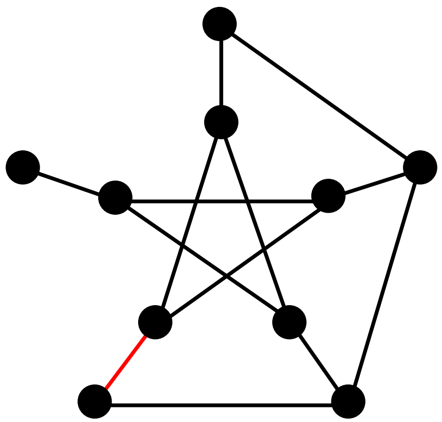
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



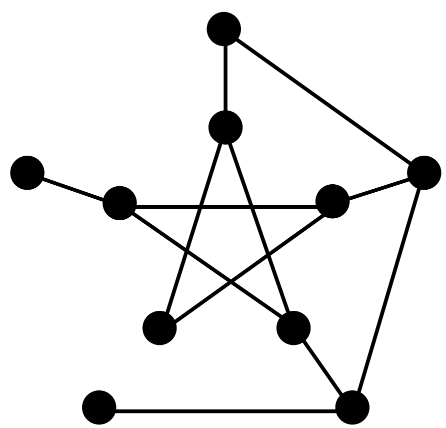
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



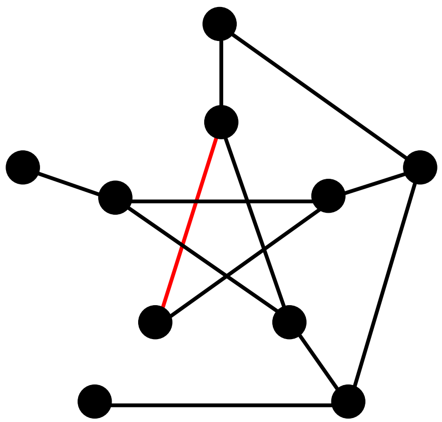
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



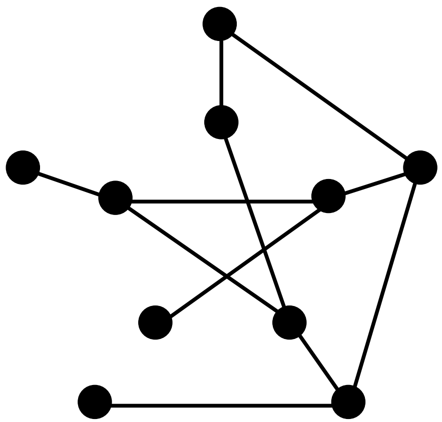
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



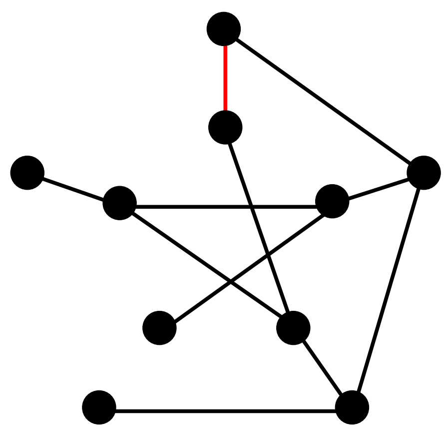
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



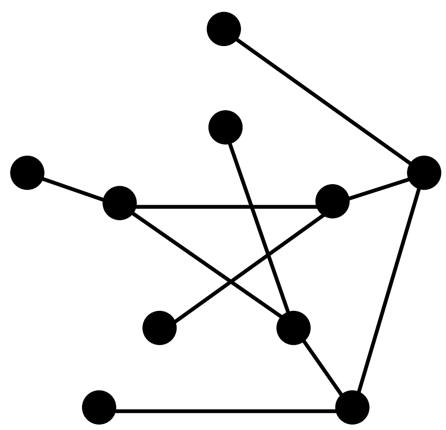
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



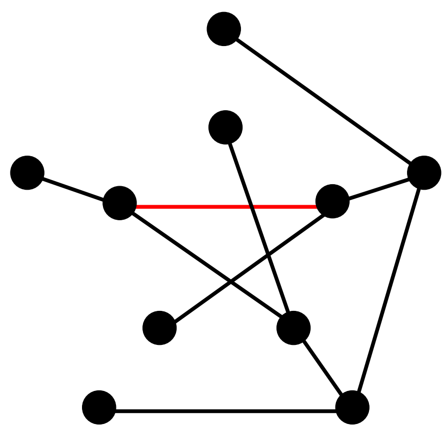
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



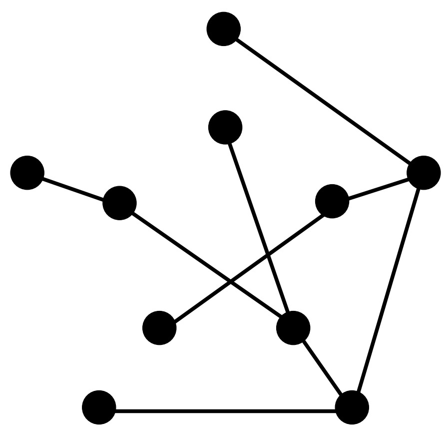
The cut-down algorithm

Given a connected graph G to construct a spanning tree:

1. If G is a tree stop.

2. Choose an edge e from a cycle and replace G with $G - e$. Repeat from 1.

Example 2.49.



The build-up algorithm

Given a connected graph G to construct a spanning tree:

The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.

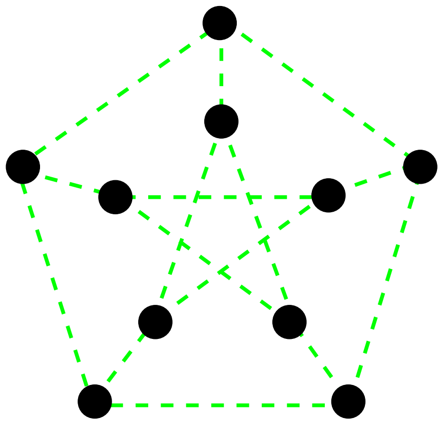
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



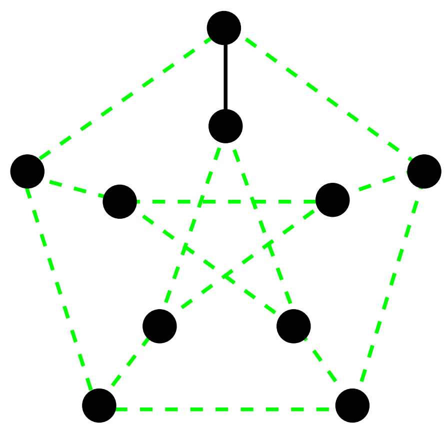
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



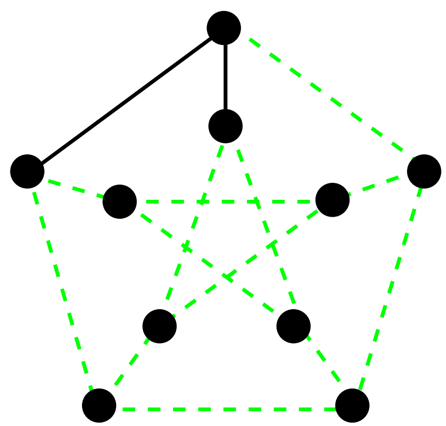
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



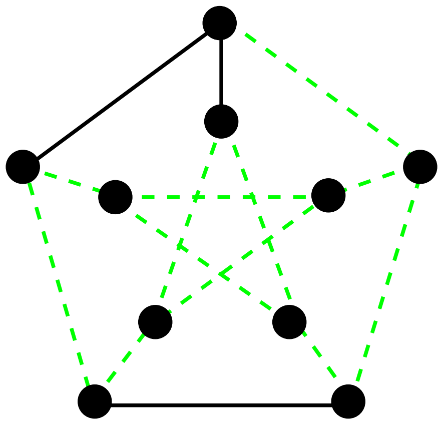
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



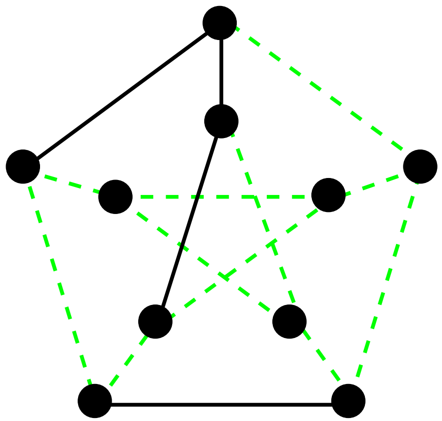
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



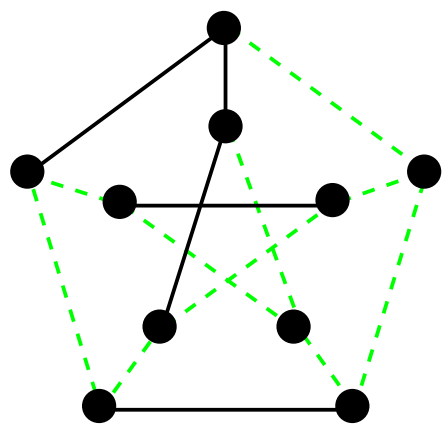
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



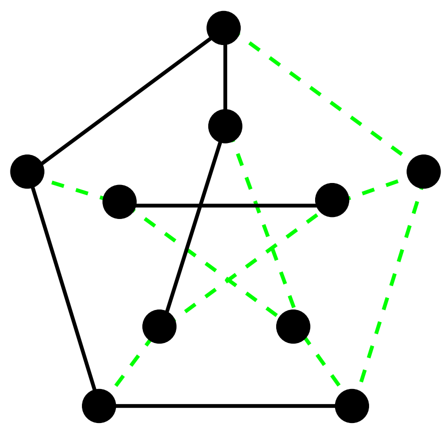
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



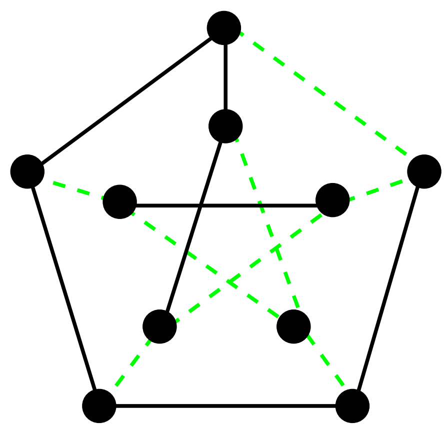
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



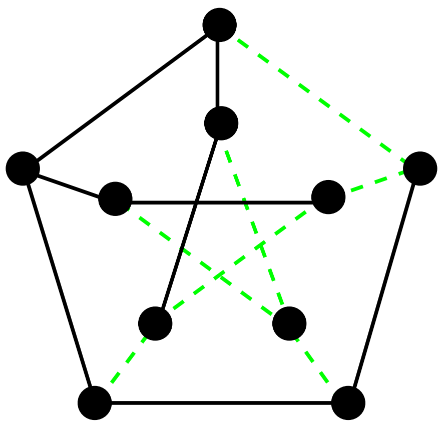
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



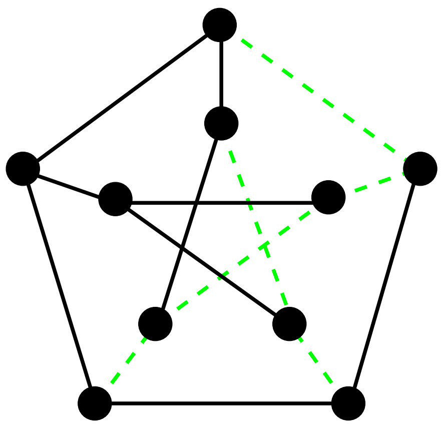
The build-up algorithm

Given a connected graph G to construct a spanning tree: start with a graph T consisting of the vertices of G and no edges.

1. If T is connected stop.

2. Add an edge e of G to T which does not form a cycle in T. Repeat from 1.

Example 2.50.



Weighted graphs

Definition 2.51. Let G be a connected graph with edge set E. To each edge $e \in E$ assign a non–negative real number $w(e)$.

Weighted graphs

Definition 2.51. Let G be a connected graph with edge set E. To each edge $e \in E$ assign a non–negative real number $w(e)$.

Then G is called a weighted graph and the number $w(e)$ is called the weight of e.

Weighted graphs

Definition 2.51. Let G be a connected graph with edge set E. To each edge $e \in E$ assign a non–negative real number $w(e)$.

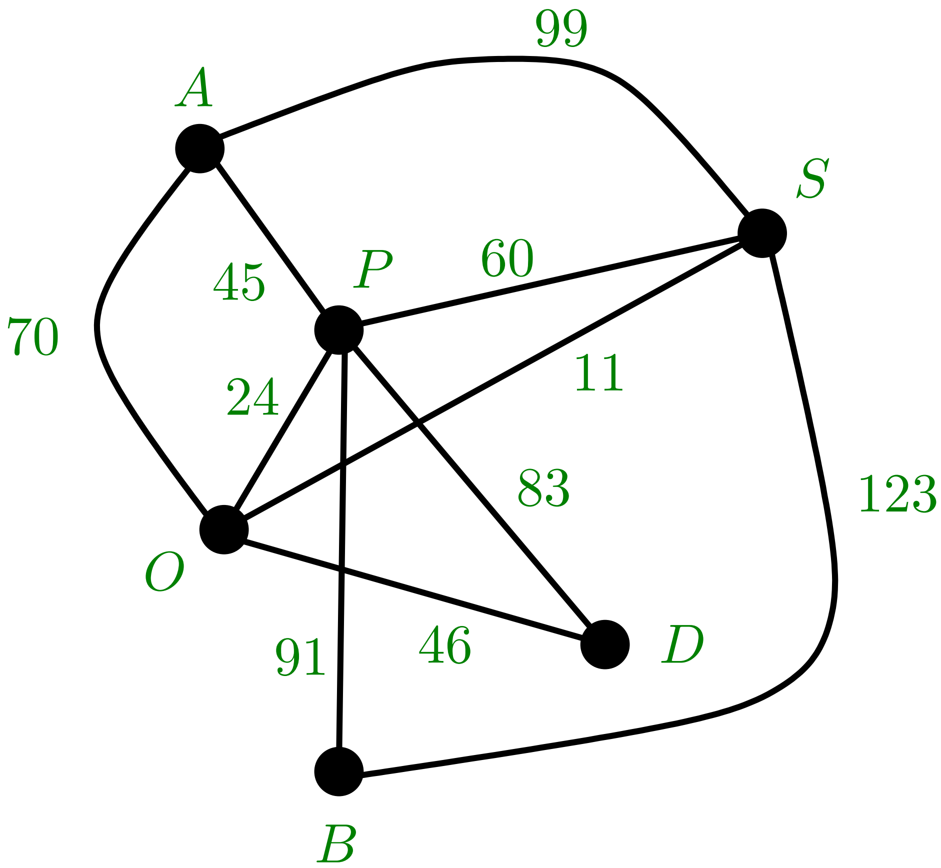
Then G is called a weighted graph and the number $w(e)$ is called the weight of e.

The sum

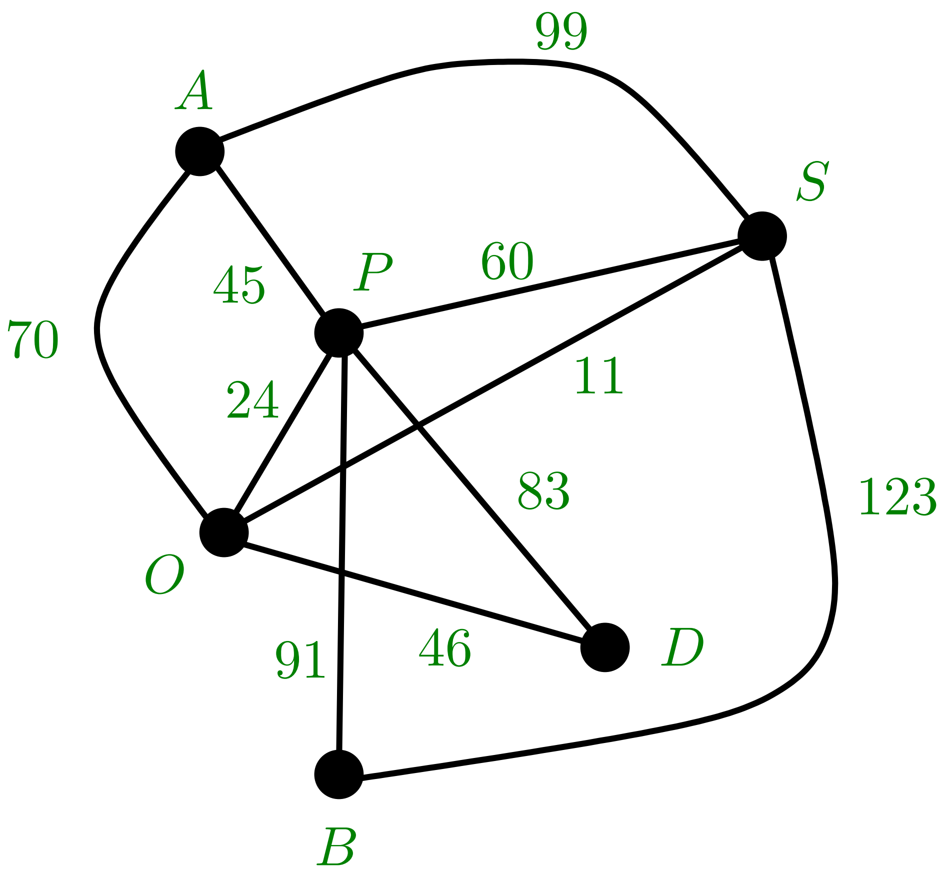
$$W(G) = \sum_{e \in E} w(e)$$

is called the weight of G.

Example 2.52.



Example 2.52.



The graph has weight $W(G) = 652$.

The Minimum Connector Problem

A subgraph of a connected graph G which contains all the vertices of G is called a spanning subgraph.

The Minimum Connector Problem

A subgraph of a connected graph G which contains all the vertices of G is called a spanning subgraph.

Every spanning graph must contain a spanning tree.

The Minimum Connector Problem

A subgraph of a connected graph G which contains all the vertices of G is called a spanning subgraph.

Every spanning graph must contain a spanning tree.

In a connected, weighted graph the problem of finding a spanning subgraph of minimal weight is called the minimal connector problem.

The Minimum Connector Problem

A subgraph of a connected graph G which contains all the vertices of G is called a spanning subgraph.

Every spanning graph must contain a spanning tree.

In a connected, weighted graph the problem of finding a spanning subgraph of minimal weight is called the minimal connector problem.

A spanning subgraph of minimal weight is always a spanning tree,

The Minimum Connector Problem

A subgraph of a connected graph G which contains all the vertices of G is called a spanning subgraph.

Every spanning graph must contain a spanning tree.

In a connected, weighted graph the problem of finding a spanning subgraph of minimal weight is called the minimal connector problem.

A spanning subgraph of minimal weight is always a spanning tree,

so the problem is to find a spanning tree of minimal weight.

The Minimum Connector Problem

A subgraph of a connected graph G which contains all the vertices of G is called a spanning subgraph.

Every spanning graph must contain a spanning tree.

In a connected, weighted graph the problem of finding a spanning subgraph of minimal weight is called the minimal connector problem.

A spanning subgraph of minimal weight is always a spanning tree,

so the problem is to find a spanning tree of minimal weight.

The following algorithm does so. Again we leave aside the problem of testing for a cycle.

The Greedy Algorithm (also known as Kruskal's Algorithm)

Let G be a connected weighted graph. To find a spanning tree T for G of minimal weight:

The Greedy Algorithm (also known as Kruskal's Algorithm)

Let G be a connected weighted graph. To find a spanning tree T for G of minimal weight:

Step 1

The Greedy Algorithm (also known as Kruskal's Algorithm)

Let G be a connected weighted graph. To find a spanning tree T for G of minimal weight:

Step 1

Step 2

The Greedy Algorithm (also known as Kruskal's Algorithm)

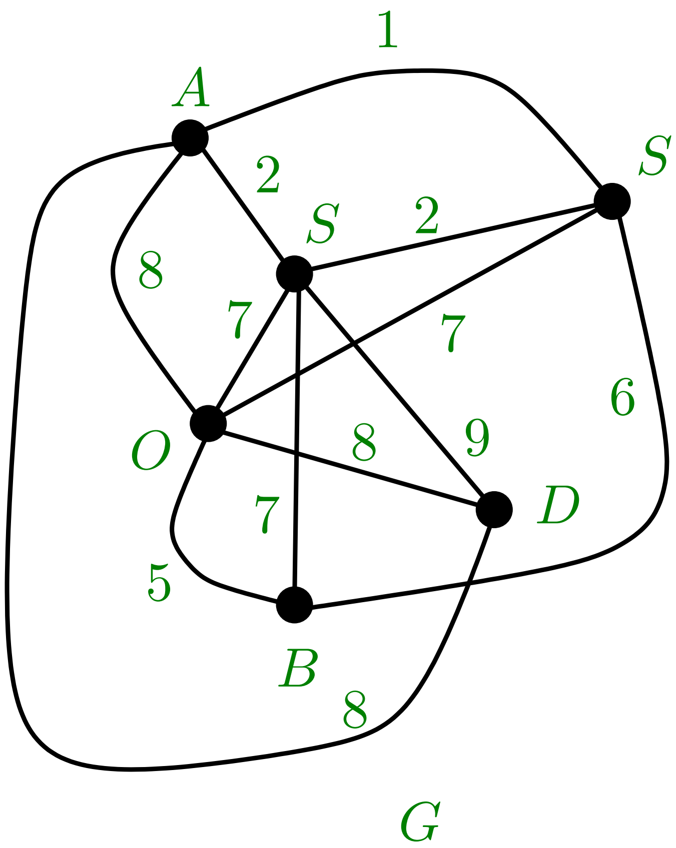
Let G be a connected weighted graph. To find a spanning tree T for G of minimal weight:

Step 1

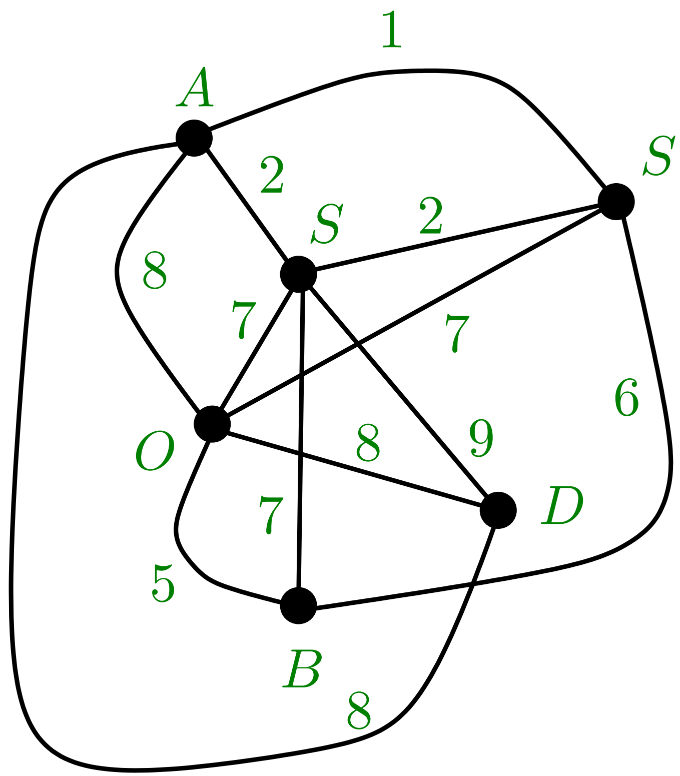
Step 2

Step 3

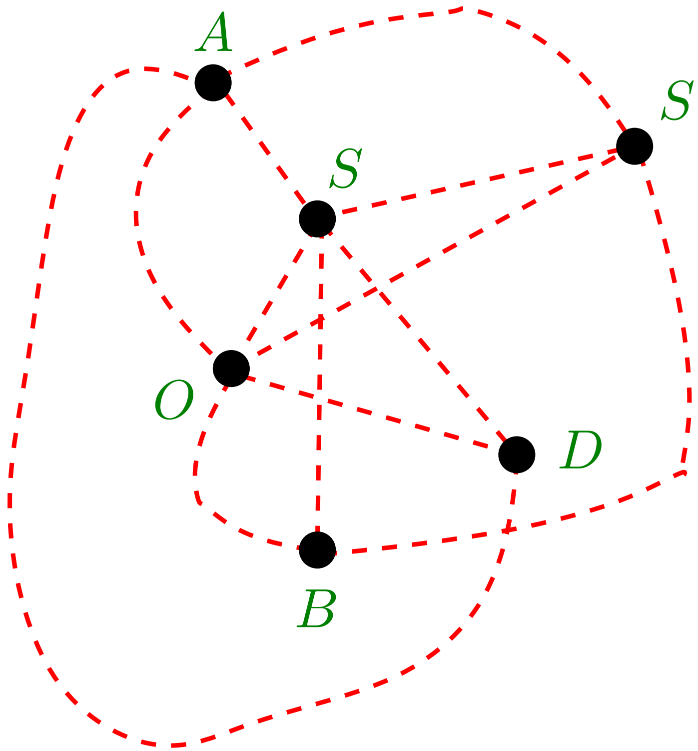
Example 2.53



Example 2.53

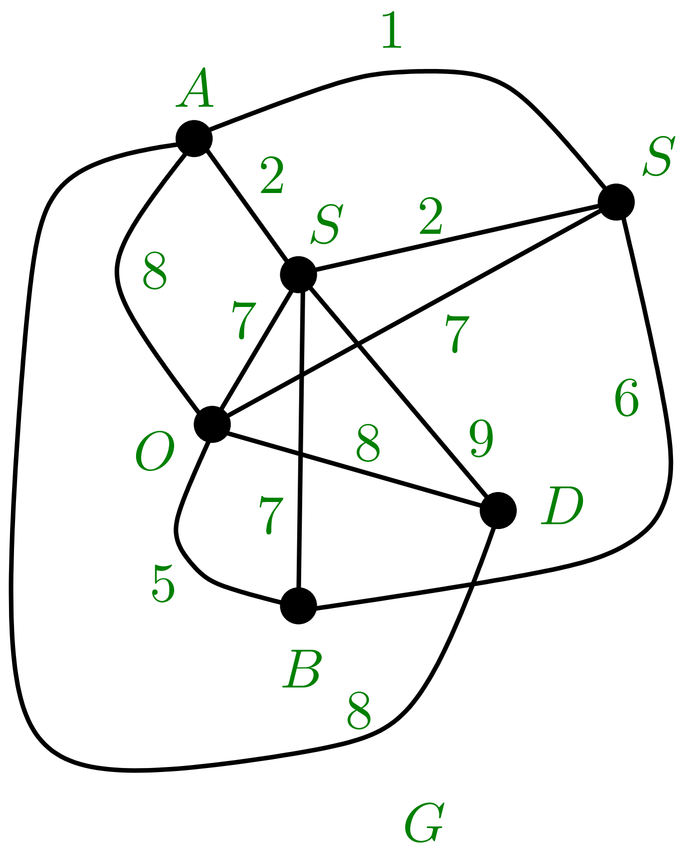


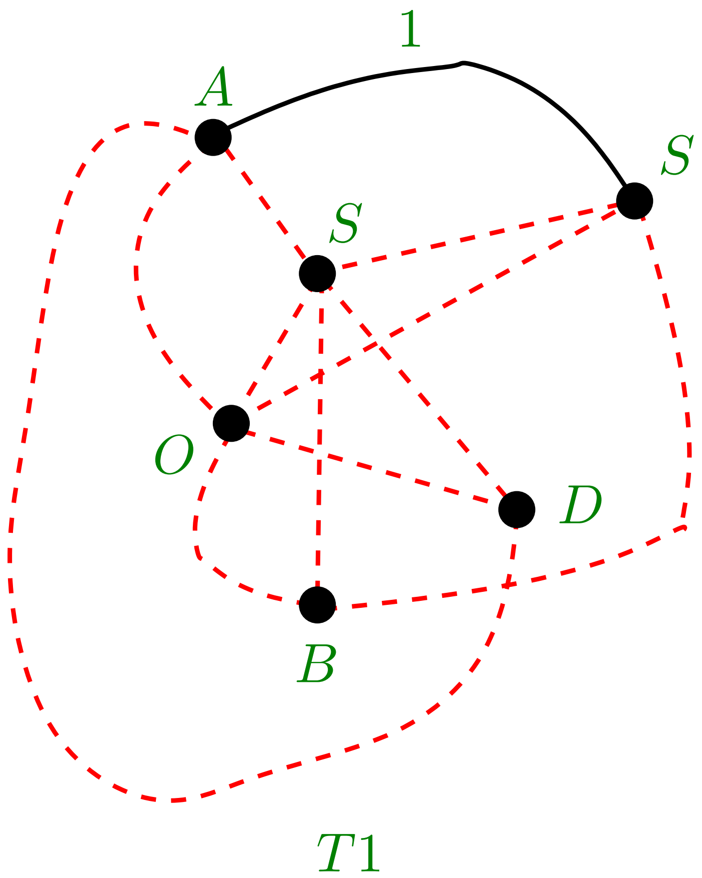
G



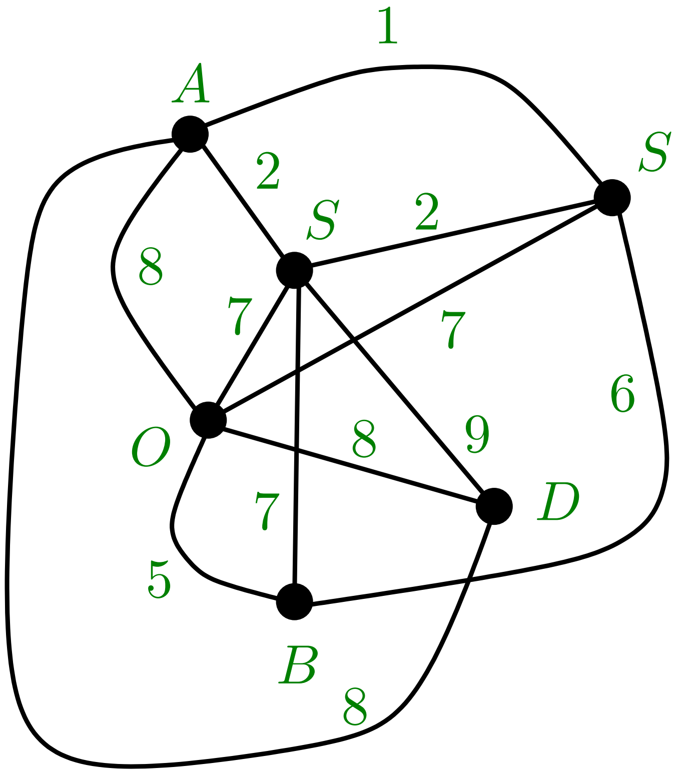
T

Example 2.53

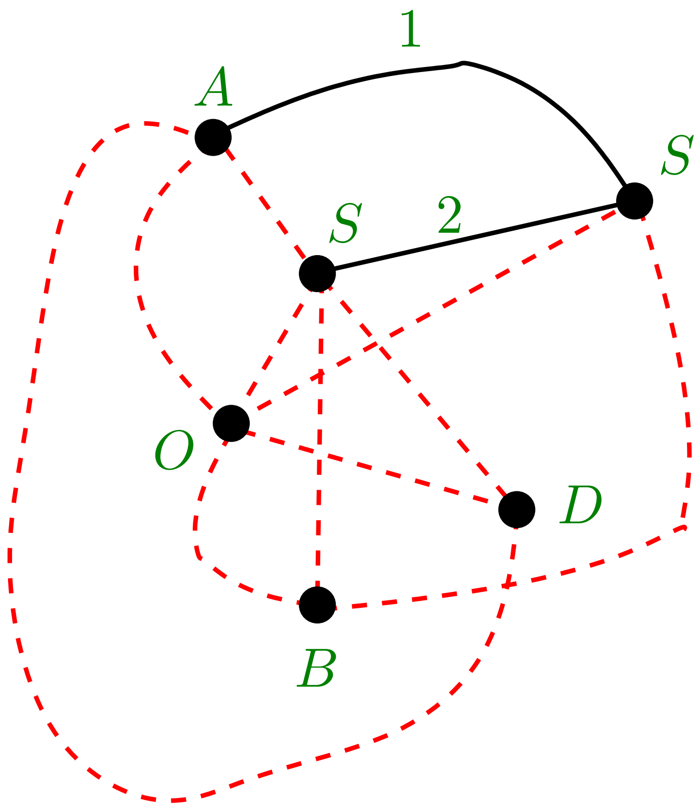




Example 2.53

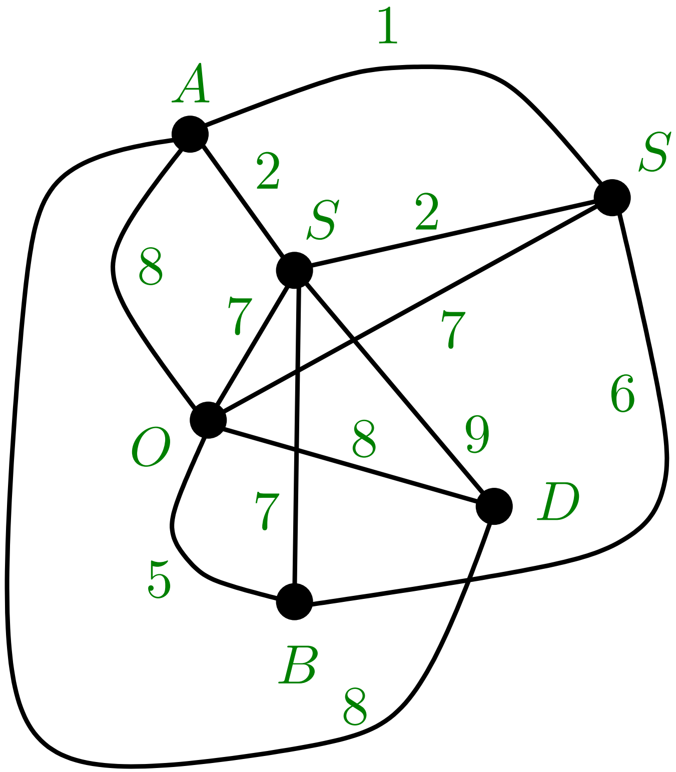


G

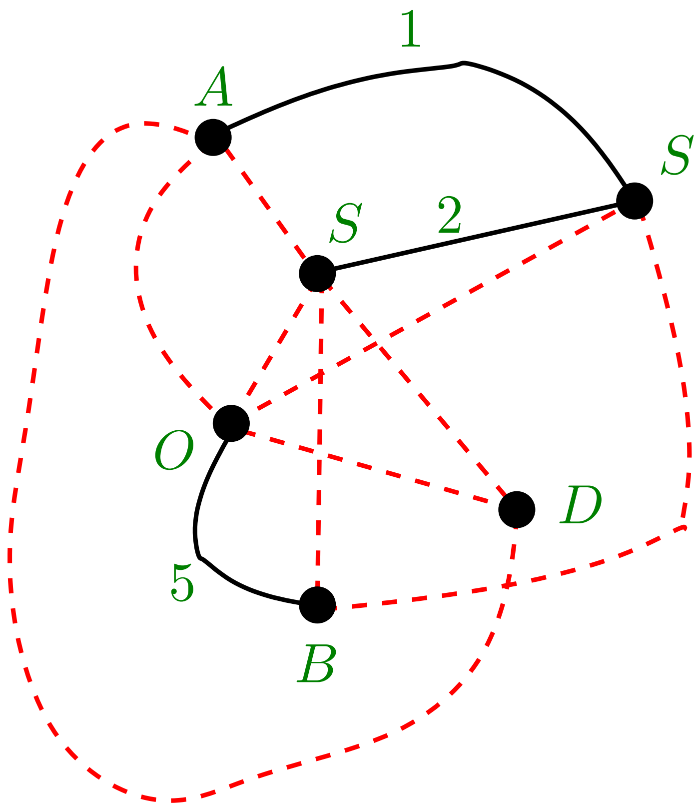


$T2$

Example 2.53

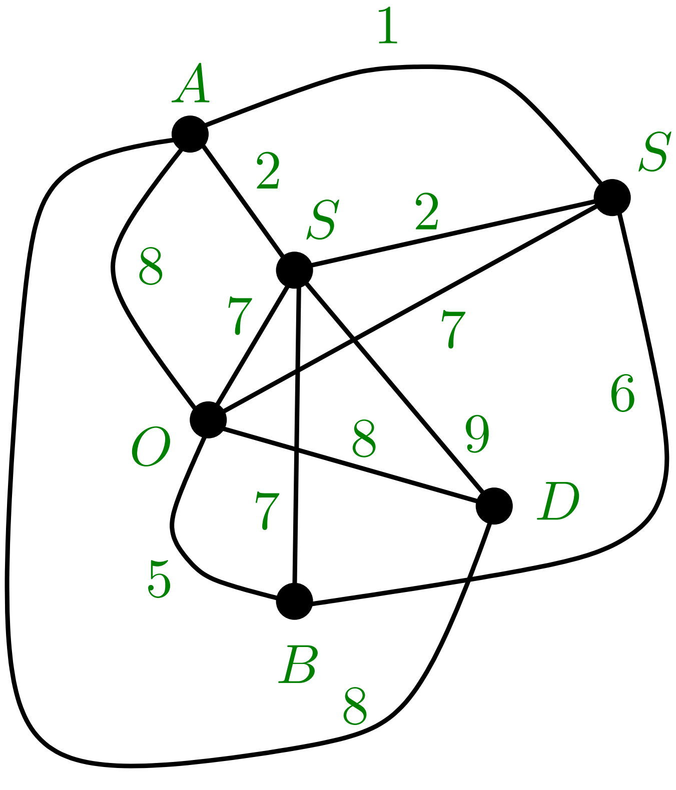


G

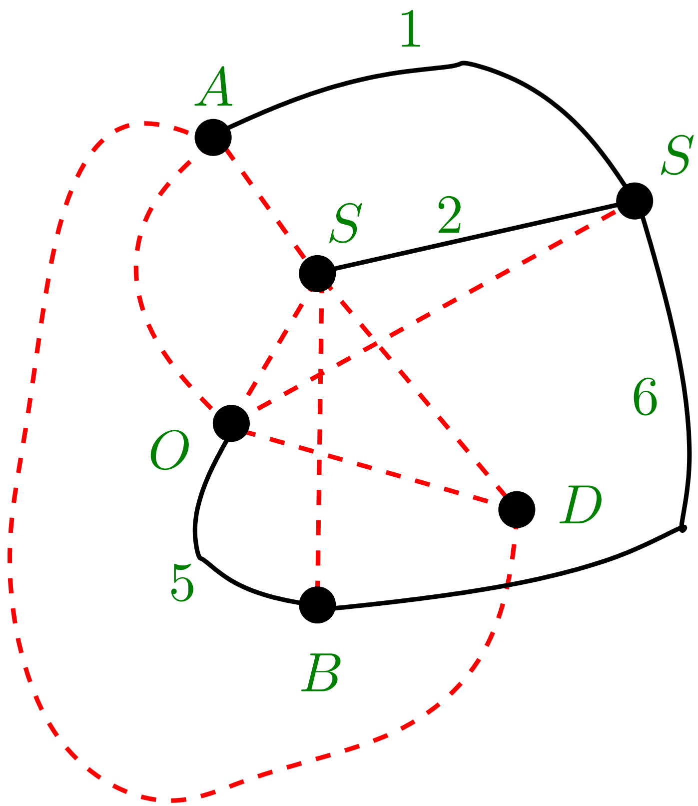


T3

Example 2.53

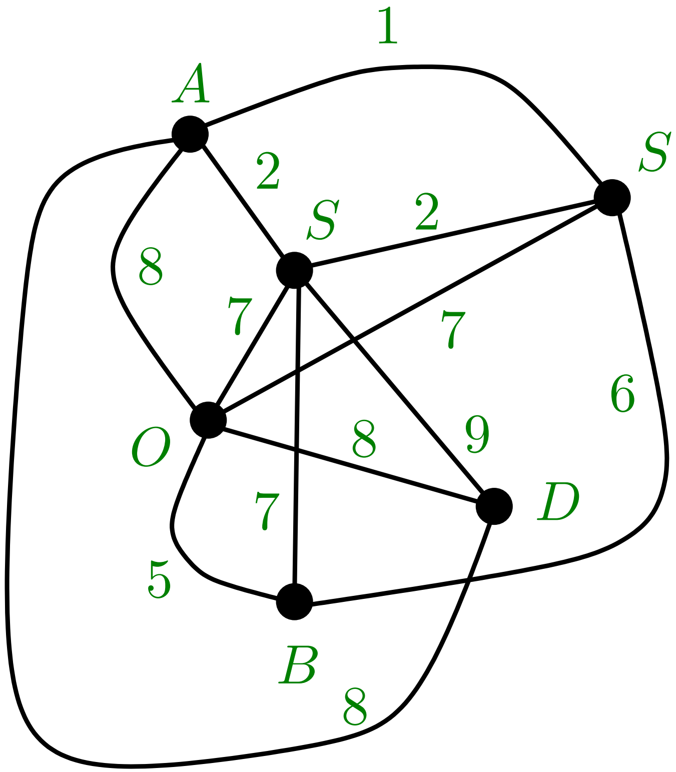


G

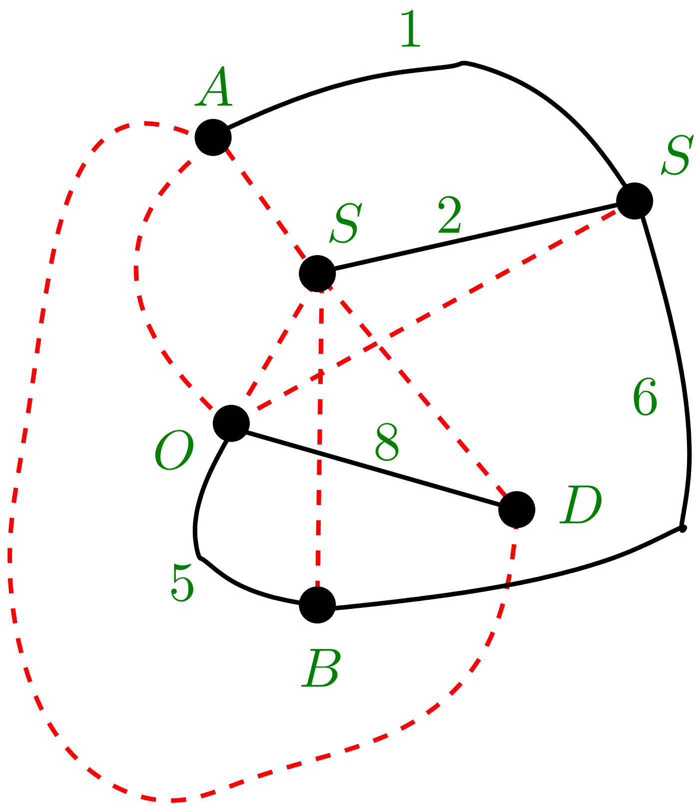


T4

Example 2.53



G



$T5$

Some choices that have to be made in the running of the algorithm.

Some choices that have to be made in the running of the algorithm.

For instance, either of the edges of weight 2 could have been included in T.

Some choices that have to be made in the running of the algorithm.

For instance, either of the edges of weight 2 could have been included in T.

A different choice results in a different minimal weight spanning tree, of which there may be many.

The Travelling Salesman Problem

A problem:

"Given a connected weighted graph G, find a closed walk in G containing all vertices of G and of minimal weight amongst all such closed walks."

This problem is very difficult to solve in general.

The Travelling Salesman Problem

A problem:

"Given a connected weighted graph G, find a closed walk in G containing all vertices of G and of minimal weight amongst all such closed walks."

This problem is very difficult to solve in general.

An easier problem: the Travelling Salesman problem:

"Given a connected weighted graph G, find a minimal weight Hamiltonian closed path in G."

The Travelling Salesman Problem

A problem:

"Given a connected weighted graph G, find a closed walk in G containing all vertices of G and of minimal weight amongst all such closed walks."

This problem is very difficult to solve in general.

An easier problem: the Travelling Salesman problem:

"Given a connected weighted graph G, find a minimal weight Hamiltonian closed path in G."

easier — fewer possible solutions,

The Travelling Salesman Problem

A problem:

"Given a connected weighted graph G, find a closed walk in G containing all vertices of G and of minimal weight amongst all such closed walks."

This problem is very difficult to solve in general.

An easier problem: the Travelling Salesman problem:

"Given a connected weighted graph G, find a minimal weight Hamiltonian closed path in G."

easier — fewer possible solutions,

but still very difficult to solve.

The Travelling Salesman Problem

A problem:

"Given a connected weighted graph G, find a closed walk in G containing all vertices of G and of minimal weight amongst all such closed walks."

This problem is very difficult to solve in general.

An easier problem: the Travelling Salesman problem:

"Given a connected weighted graph G, find a minimal weight Hamiltonian closed path in G."

easier — fewer possible solutions,

but still very difficult to solve.

The algorithm for the Minimum Connector problem can be used to find a lower bound for the Travelling Salesman problem.

Deleting vertices

Definition 2.54. Let G be a graph and let v be a vertex of G.

Deleting vertices

Definition 2.54. Let G be a graph and let v be a vertex of G.

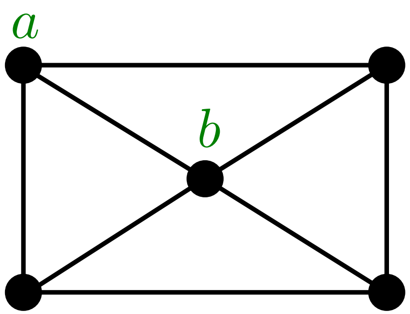
The graph $G - v$ obtained from G by deleting v is defined to be the graph formed by removing v and all its incident edges from G.

Deleting vertices

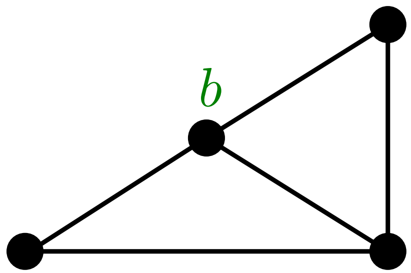
Definition 2.54. Let G be a graph and let v be a vertex of G.

The graph $G - v$ obtained from G by deleting v is defined to be the graph formed by removing v and all its incident edges from G.

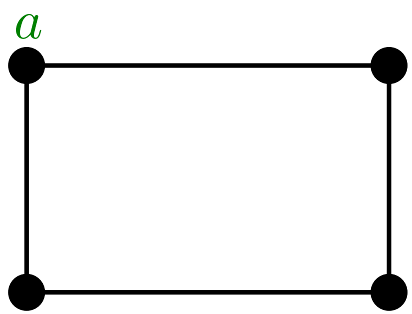
Example 2.55.



G



$G - a$



$G - b$

A lower bound for the Travelling Salesman

Theorem 2.56. *If G is a weighted graph, C is a minimal weight Hamiltonian closed path in G and v is a vertex of G then*

$$w(C) \geq M + m_1 + m_2,$$

where M is the weight of a minimal weight spanning tree for $G - v$ and m_1 and m_2 are the weights of two edges of least weight incident to v.

A lower bound for the Travelling Salesman

Theorem 2.56. *If G is a weighted graph, C is a minimal weight Hamiltonian closed path in G and v is a vertex of G then*

$$w(C) \geq M + m_1 + m_2,$$

where M is the weight of a minimal weight spanning tree for $G - v$ and m_1 and m_2 are the weights of two edges of least weight incident to v.

As pointed out above the inequality in this Theorem may be strict.

A lower bound for the Travelling Salesman

Theorem 2.56. *If G is a weighted graph, C is a minimal weight Hamiltonian closed path in G and v is a vertex of G then*

$$w(C) \geq M + m_1 + m_2,$$

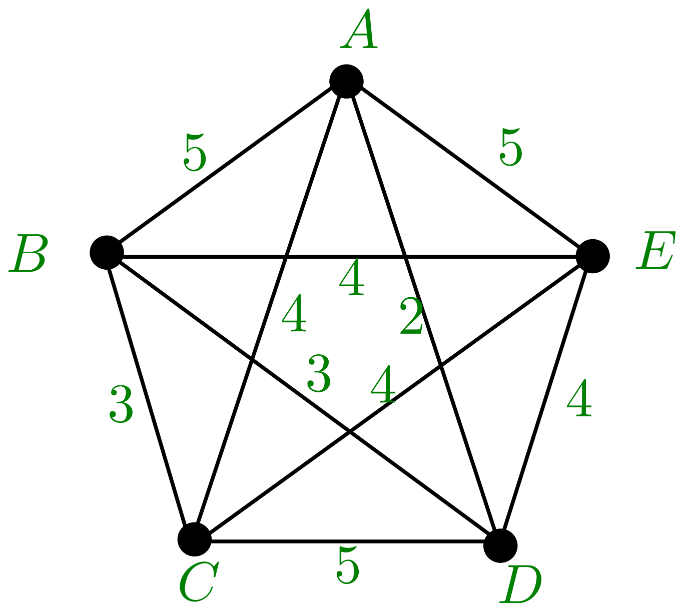
where M is the weight of a minimal weight spanning tree for $G - v$ and m_1 and m_2 are the weights of two edges of least weight incident to v.

As pointed out above the inequality in this Theorem may be strict.

We obtain a lower bound for the Travelling Salesman problem, which in some cases may be smaller than the weight of minimal weight Hamiltonian closed path.

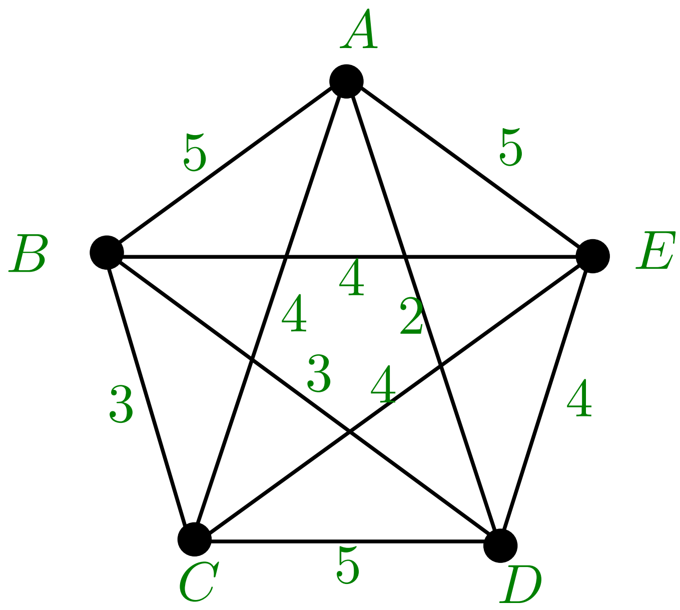
Example 2.57.

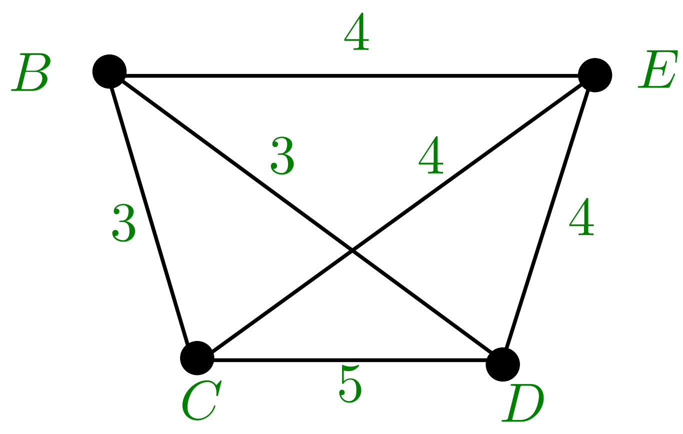
Find a lower bound for the Travelling salesman problem in the weighted graph G below by removing vertex A.



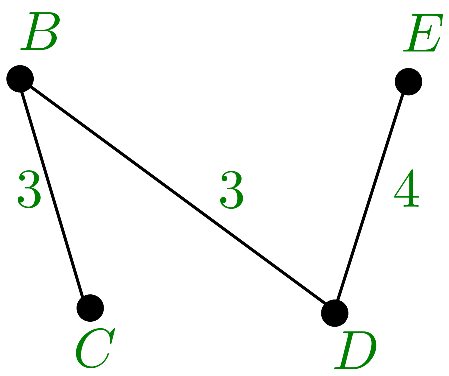
Example 2.57.

Find a lower bound for the Travelling salesman problem in the weighted graph G below by removing vertex A.

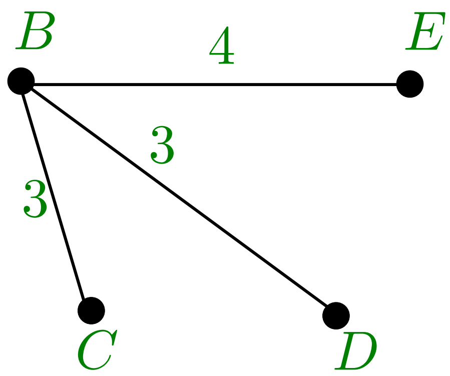




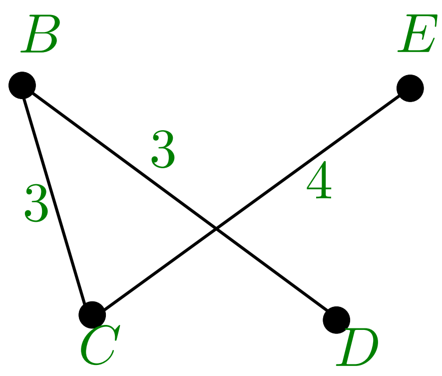
The Greedy Algorithm output: one of the 3 trees shown below, all of weight 10; so $M = 10$.



Spanning Tree 1

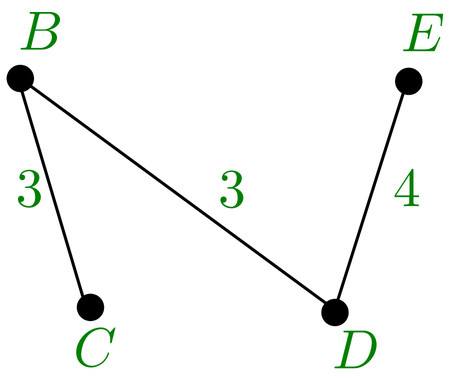


Spanning Tree 2

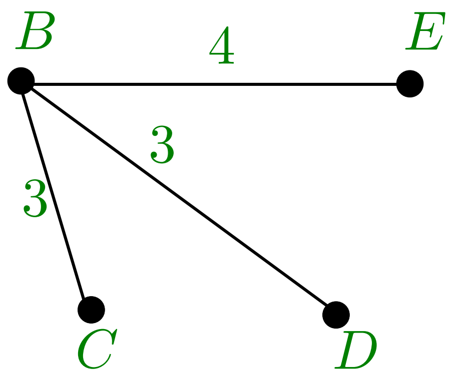


Spanning Tree 3

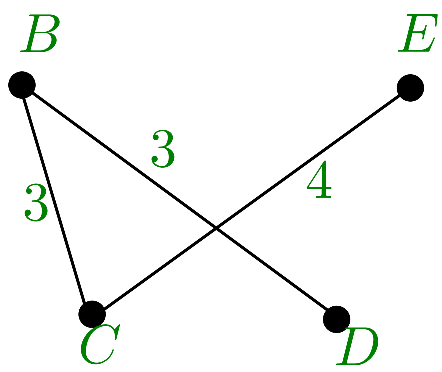
The Greedy Algorithm output: one of the 3 trees shown below, all of weight 10; so $M = 10$.



Spanning Tree 1



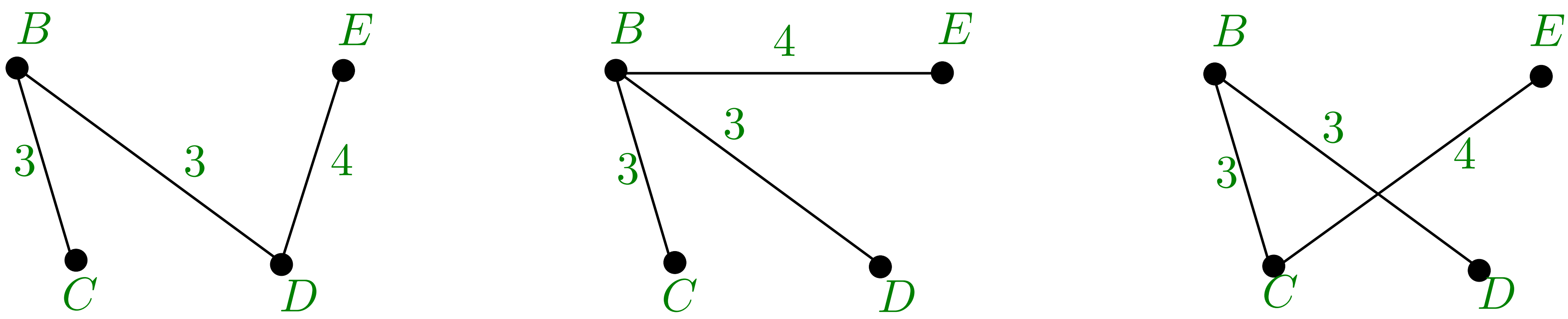
Spanning Tree 2



Spanning Tree 3

Edges of minimal weight incident to A: $\{A, C\}$ and $\{A, D\}$ of weights $m_1 = 2$ and $m_2 = 4$.

The Greedy Algorithm output: one of the 3 trees shown below, all of weight 10; so $M = 10$.

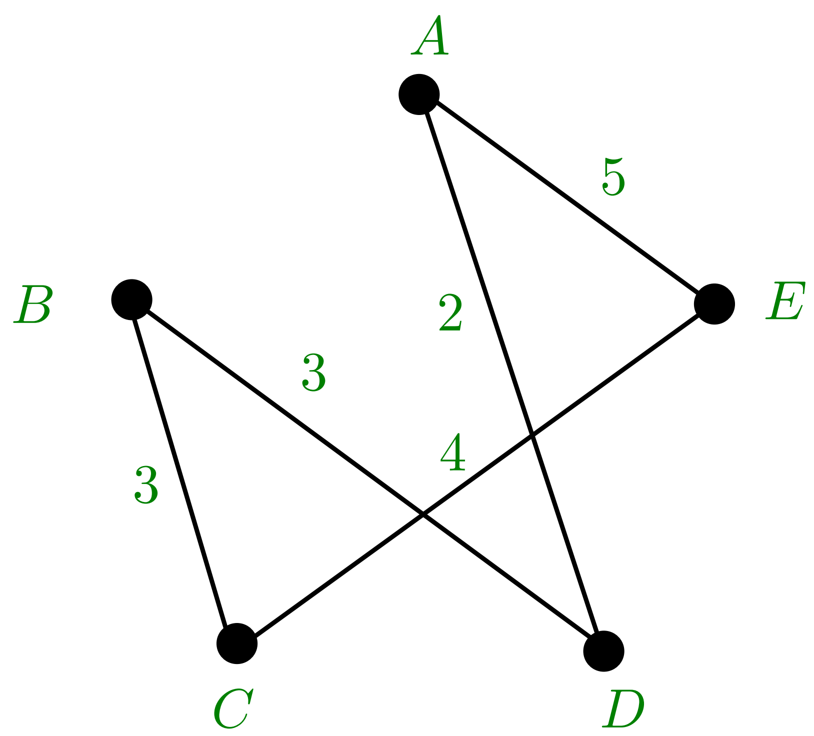


Spanning Tree 1 Spanning Tree 2 Spanning Tree 3

Edges of minimal weight incident to A: $\{A, C\}$ and $\{A, D\}$ of weights $m_1 = 2$ and $m_2 = 4$.

Lower bound $10 + 2 + 4 = 16$.

A minimal weight Hamiltonian closed path



An electronics engineer wishes to make a board on which there are 3 input terminals and 3 output terminals.

An electronics engineer wishes to make a board on which there are 3 input terminals and 3 output terminals.

Each input terminal is to be connected to all output terminals.

An electronics engineer wishes to make a board on which there are 3 input terminals and 3 output terminals.

Each input terminal is to be connected to all output terminals.

Connections are to be made by lines of solder laid on the board (not necessarily straight).

An electronics engineer wishes to make a board on which there are 3 input terminals and 3 output terminals.

Each input terminal is to be connected to all output terminals.

Connections are to be made by lines of solder laid on the board (not necessarily straight).

Two different lines of solder must not cross.

An electronics engineer wishes to make a board on which there are 3 input terminals and 3 output terminals.

Each input terminal is to be connected to all output terminals.

Connections are to be made by lines of solder laid on the board (not necessarily straight).

Two different lines of solder must not cross.

Is this possible?

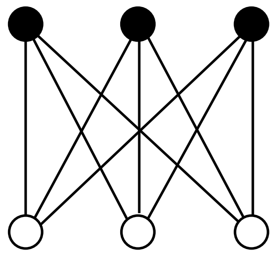
An electronics engineer wishes to make a board on which there are 3 input terminals and 3 output terminals.

Each input terminal is to be connected to all output terminals.

Connections are to be made by lines of solder laid on the board (not necessarily straight).

Two different lines of solder must not cross.

Is this possible?



The complete bipartite graph $K_{3,3}$

Planar Graphs

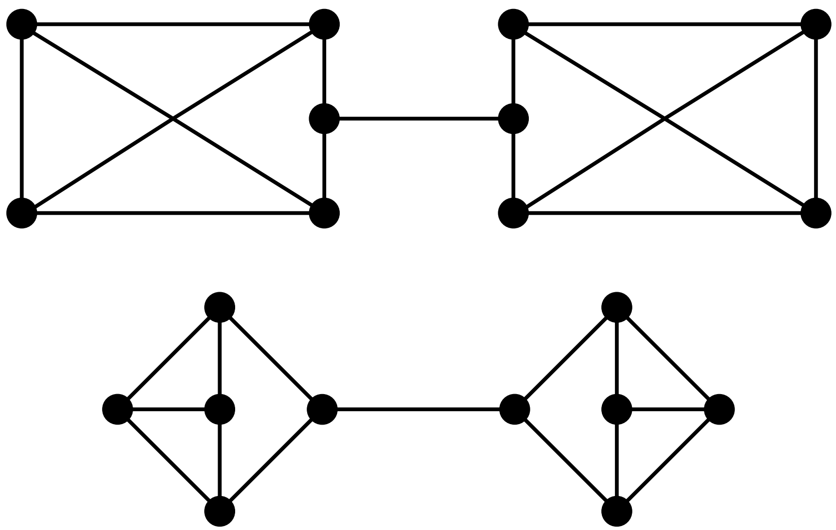
Definition 2.58. A graph is planar if it can be drawn in the plane without edges crossing.

Planar Graphs

Definition 2.58. A graph is planar if it can be drawn in the plane without edges crossing.

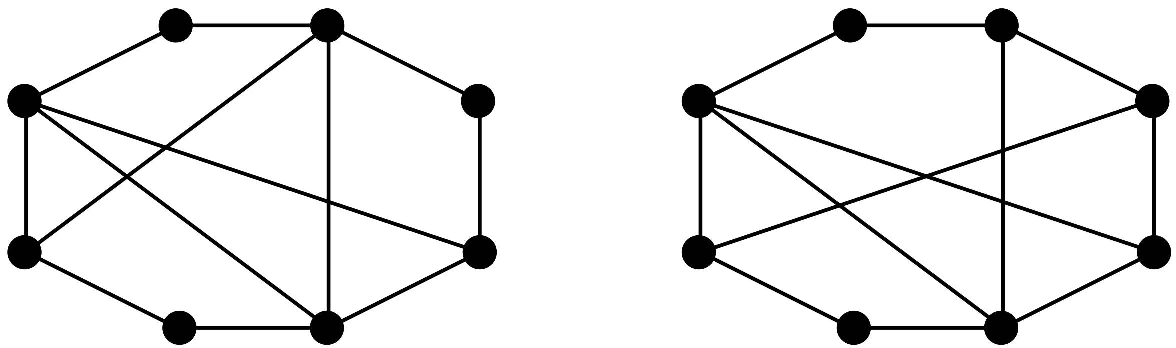
Example 2.59.

1.



Example 2.59 cont.

2.



Faces

Definition 2.60. Let D be a planar graph, drawn on the plane.

Faces

Definition 2.60. Let D be a planar graph, drawn on the plane.

If x is a point of the plane not lying on D then the set of all points of the plane that can be reached from x without crossing D is called a face of D.

Faces

Definition 2.60. Let D be a planar graph, drawn on the plane.

If x is a point of the plane not lying on D then the set of all points of the plane that can be reached from x without crossing D is called a face of D.

One face is always unbounded and is called the exterior face.

Faces

Definition 2.60. Let D be a planar graph, drawn on the plane.

If x is a point of the plane not lying on D then the set of all points of the plane that can be reached from x without crossing D is called a face of D.

One face is always unbounded and is called the exterior face.

(To make a rigourous definition of face requires the Jordan Curve theorem, which says that:

a simple closed curve in the plane divides the plane into two parts, one inside and one outside the curve.

Faces

Definition 2.60. Let D be a planar graph, drawn on the plane.

If x is a point of the plane not lying on D then the set of all points of the plane that can be reached from x without crossing D is called a face of D.

One face is always unbounded and is called the exterior face.

(To make a rigourous definition of face requires the Jordan Curve theorem, which says that:

a simple closed curve in the plane divides the plane into two parts, one inside and one outside the curve.

This theorem is beyond the scope of this course.)

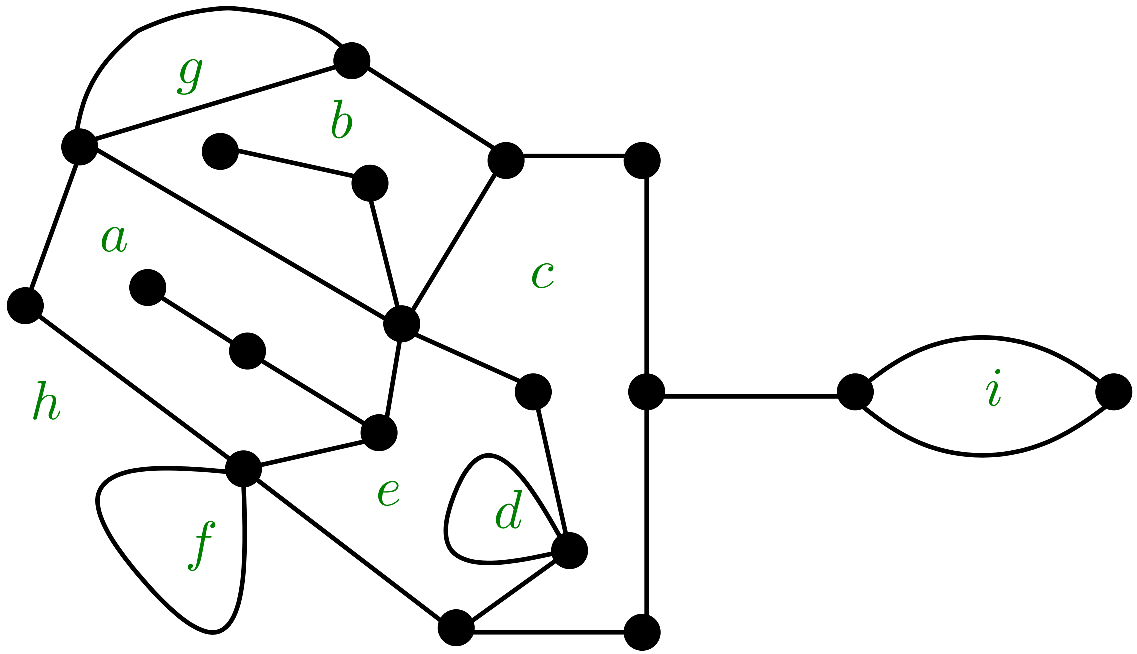
Example 2.61.

1. All trees are planar and have one face (which is exterior).

Example 2.61.

1. All trees are planar and have one face (which is exterior).

2. The graph below has 9 faces labelled $a \ldots i$. Face h is the exterior face.



Euler's Formula

Theorem 2.62. *[Euler's Formula] Let G be a connected planar graph (drawn in the plane) with n vertices, m edges and r faces. Then $n - m + r = 2$.*

Euler's Formula

Theorem 2.62. *[Euler's Formula] Let* G *be a connected planar graph (drawn in the plane) with* n *vertices,* m *edges and* r *faces. Then* $n - m + r = 2$*.*

Definition 2.63. Let F be a face of a planar graph. The degree of F, denoted $\deg(F)$ is the number of edges in the boundary of F, where edges lying in no face except F count twice.

Euler's Formula

Theorem 2.62. *[Euler's Formula] Let G be a connected planar graph (drawn in the plane) with n vertices, m edges and r faces. Then $n - m + r = 2$.*

Definition 2.63. Let F be a face of a planar graph. The degree of F, denoted $\deg(F)$ is the number of edges in the boundary of F, where edges lying in no face except F count twice.

(To compute $\deg(F)$ walk once round the boundary of F, counting each edge on the way.)

Non-planarity

Lemma 2.64. *If* G *is a planar graph with* m *edges and* r *faces* $F_1, \ldots, F_r$ *then*

$$\sum_{i=1}^{r} \deg(F_i) = 2m.$$

Non-planarity

Lemma 2.64. *If G is a planar graph with m edges and r faces $F_1, \ldots, F_r$ then*

$$\sum_{i=1}^{r} \deg(F_i) = 2m.$$

Corollary 2.65. *If G is a simple connected planar graph with $n \geq 3$ vertices and m edges then $m \leq 3n - 6$.*

Non-planarity

Lemma 2.64. *If G is a planar graph with m edges and r faces $F_1, \ldots, F_r$ then*

$$\sum_{i=1}^{r} \deg(F_i) = 2m.$$

Corollary 2.65. *If G is a simple connected planar graph with $n \geq 3$ vertices and m edges then $m \leq 3n - 6$.*

Corollary 2.66. *If G is a connected simple planar graph with $n \geq 3$ vertices, m edges and no cycle of length 3 then $m \leq 2n - 4$.*

Non-planarity

Lemma 2.64. *If G is a planar graph with m edges and r faces $F_1, \ldots, F_r$ then*

$$\sum_{i=1}^{r} \deg(F_i) = 2m.$$

Corollary 2.65. *If G is a simple connected planar graph with $n \geq 3$ vertices and m edges then $m \leq 3n - 6$.*

Corollary 2.66. *If G is a connected simple planar graph with $n \geq 3$ vertices, m edges and no cycle of length 3 then $m \leq 2n - 4$.*

Theorem 2.67. *The complete graph K_5 and the complete bipartite graph $K_{3,3}$ are both non–planar.*

Subdivision

If a graph G is non-planar then any graph which contains G as a subgraph is also non–planar.

Subdivision

If a graph G is non-planar then any graph which contains G as a subgraph is also non–planar.

It follows that if a graph contains K_5 or $K_{3,3}$ as a subgraph it must be non-planar.

Subdivision

If a graph G is non-planar then any graph which contains G as a subgraph is also non–planar.

It follows that if a graph contains K_5 or $K_{3,3}$ as a subgraph it must be non-planar.

Definition 2.68. A graph H is a subdivision of a graph G if H is obtained from G by the addition of a finite number of vertices of degree 2 to edges of G.

Subdivision

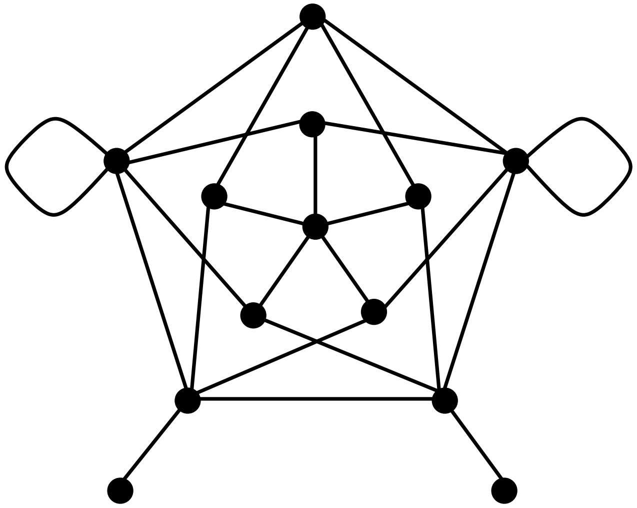
If a graph G is non-planar then any graph which contains G as a subgraph is also non–planar.

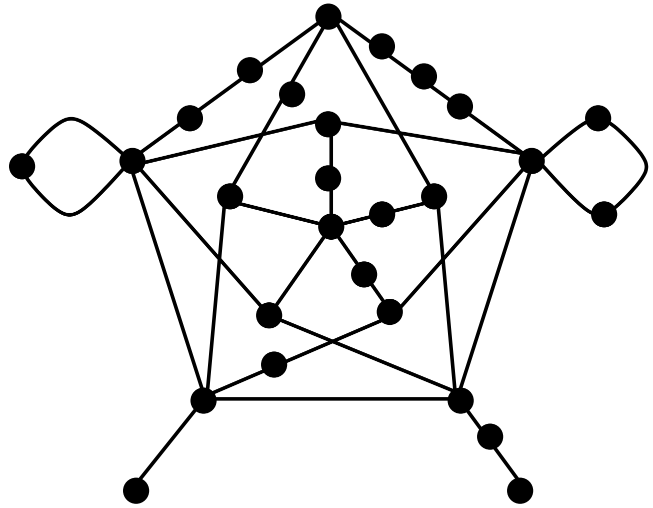
It follows that if a graph contains K_5 or $K_{3,3}$ as a subgraph it must be non-planar.

Definition 2.68. A graph H is a subdivision of a graph G if H is obtained from G by the addition of a finite number of vertices of degree 2 to edges of G.

It is possible to add no vertices and so a graph is a subdivision of itself.

Example 2.69. H below is a subdivision of G.





Planarity and subdivisions

The following theorem is an easy consequence of Theorem 2.67.

Planarity and subdivisions

The following theorem is an easy consequence of Theorem 2.67.

Theorem 2.70. *If G is a graph containing a subgraph which is a subdivision of K_5 or $K_{3,3}$ then G is non–planar.*

Example 2.71.

Neither Corollary 2.65 nor Corollary 2.66 are sufficient to show that the graphs of this example are non–planar.

Example 2.71.

Neither Corollary 2.65 nor Corollary 2.66 are sufficient to show that the graphs of this example are non–planar.

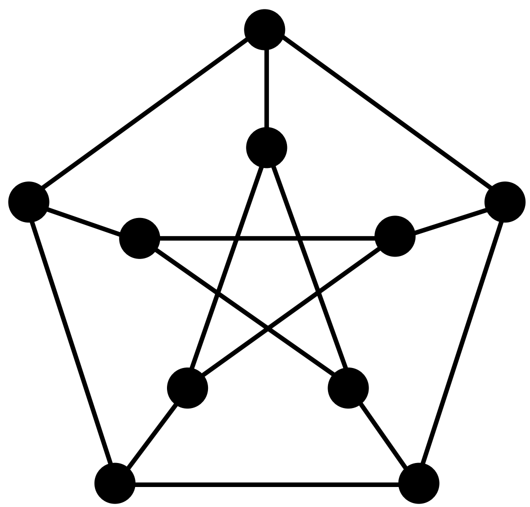
1. The Petersen graph has 10 vertices and 15 edges.

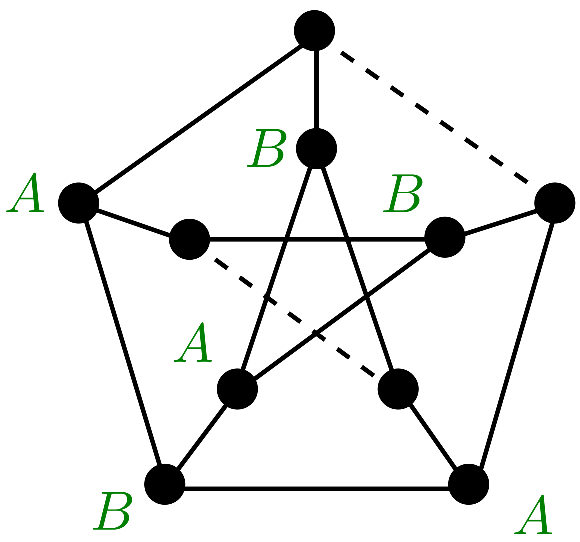
Example 2.71.

Neither Corollary 2.65 nor Corollary 2.66 are sufficient to show that the graphs of this example are non–planar.

1. The Petersen graph has 10 vertices and 15 edges.

The diagram on the right shows a subgraph which is a subdivision of $K_{3,3}$. Therefore the graph is non-planar. (Vertices which are not labelled A or B are those added in the subdivision.





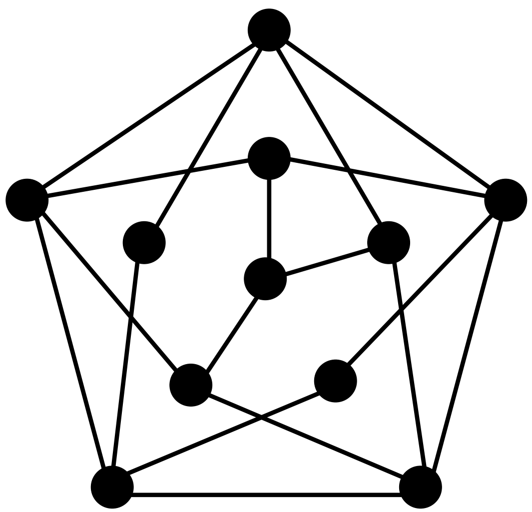
Example 2.71 cont.

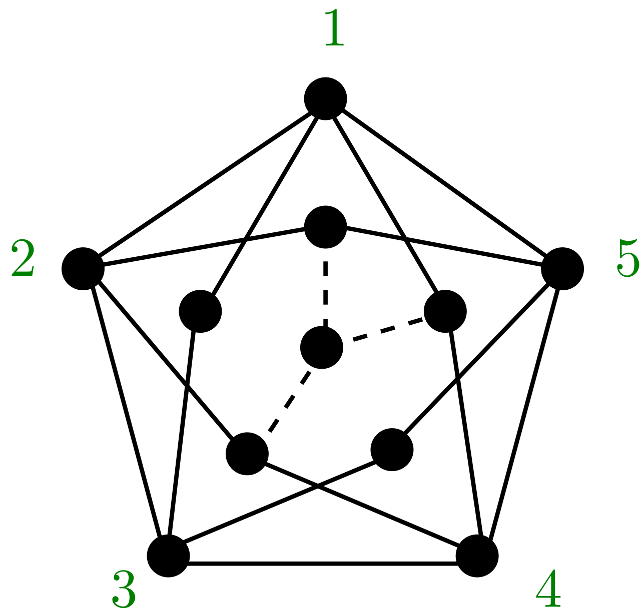
2. The graph shown below has 11 vertices and 18 edges.

Example 2.71 cont.

2. The graph shown below has 11 vertices and 18 edges.

The right hand diagram shows a subgraph which is a subdivision of K_5. Therefore the graph is non-planar.





Kuratowski's Theorem

A more surprising theorem:

Theorem 2.72. *[Kuratowski] If* G *is a non–planar graph then* G *contains a subgraph which is a subdivision of* K_5 *or* $K_{3,3}$*.*

The Four-Colour Problem

De Morgan's conjecture (1852): any map of countries can be coloured using only 4 colours, in such a way that countries with a common border have different colours.

The Four-Colour Problem

De Morgan's conjecture (1852): any map of countries can be coloured using only 4 colours, in such a way that countries with a common border have different colours.

Given a map of countries construct a planar graph as follows.

Place one vertex in each country.

The Four-Colour Problem

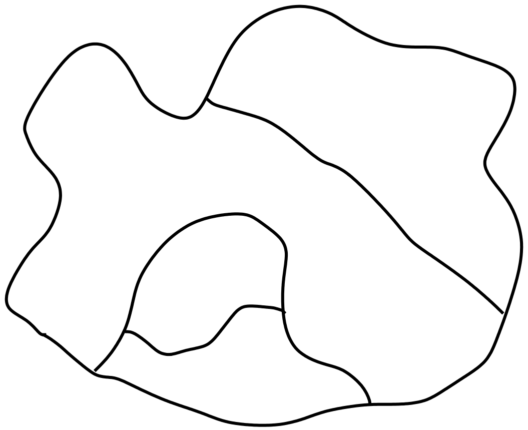
De Morgan's conjecture (1852): any map of countries can be coloured using only 4 colours, in such a way that countries with a common border have different colours.

Given a map of countries construct a planar graph as follows.

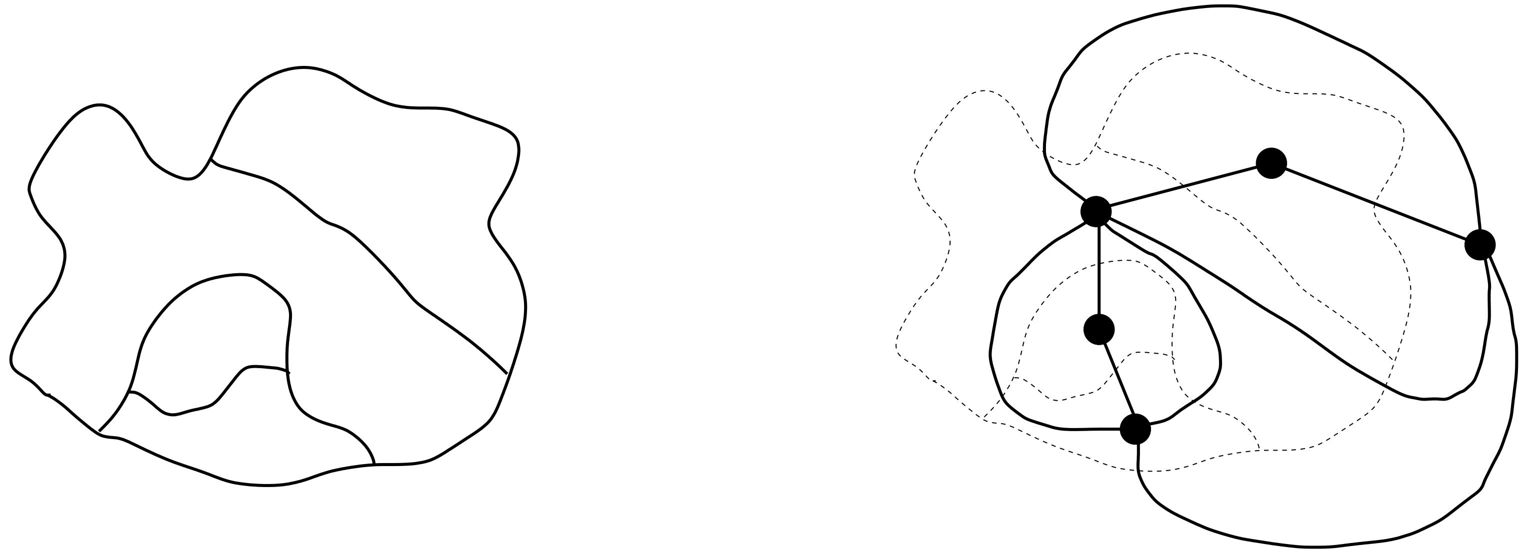
Place one vertex in each country.

Join two vertices with an edge whenever their countries have a common border.

Example 2.73.



Example 2.73.



Reformulated in terms of graph theory:

A colouring of a map according to de Morgan's rules corresponds to colouring each vertex of the graph in such a way that no two adjacent vertices have the same colour.

Reformulated in terms of graph theory:

A colouring of a map according to de Morgan's rules corresponds to colouring each vertex of the graph in such a way that no two adjacent vertices have the same colour.

De Morgan's conjecture is that this can be done with 4 colours:

Reformulated in terms of graph theory:

A colouring of a map according to de Morgan's rules corresponds to colouring each vertex of the graph in such a way that no two adjacent vertices have the same colour.

De Morgan's conjecture is that this can be done with 4 colours:

that is any such graph has a 4-colouring.

If it can be shown that any planar graph without loops is 4-colourable then it follows that every map of countries can be coloured as required.

Reformulated in terms of graph theory:

A colouring of a map according to de Morgan's rules corresponds to colouring each vertex of the graph in such a way that no two adjacent vertices have the same colour.

De Morgan's conjecture is that this can be done with 4 colours:

that is any such graph has a 4-colouring.

> If it can be shown that any planar graph without loops is 4-colourable then it follows that every map of countries can be coloured as required.

Conjecture 2.74. *[The 4–colour conjecture] Every simple planar graph is 4–colourable.*

History

1852 Guthrie and de Morgan proposed the 4–colour conjecture.

History

1852 Guthrie and de Morgan proposed the 4–colour conjecture.

1873 Cayley presented a proof to the London Mathematical Society. The proof was fatally flawed.

History

1852 Guthrie and de Morgan proposed the 4–colour conjecture.

1873 Cayley presented a proof to the London Mathematical Society. The proof was fatally flawed.

1879 Kempe published a proof; which collapsed.

History

1852 Guthrie and de Morgan proposed the 4–colour conjecture.

1873 Cayley presented a proof to the London Mathematical Society. The proof was fatally flawed.

1879 Kempe published a proof; which collapsed.

1880 Tait gave a proof which turned out (after ten years) to be incomplete.

1976 Appel & Haken at the University of Illinois proved the 4–colour conjecture. Their proof required checking $1,476$ "reducible" configurations and this used hundreds of hours of CPU time on a Cray computer.

1976 Appel & Haken at the University of Illinois proved the 4–colour conjecture. Their proof required checking $1,476$ “reducible” configurations and this used hundreds of hours of CPU time on a Cray computer.

A problem with Appel & Haken’s proof is that the program ran for so long that it is impossible to verify manually. We cannot even to be sure that the hardware performed well enough, over such an extended period, to give a reliable result.

1976 Appel & Haken at the University of Illinois proved the 4–colour conjecture. Their proof required checking $1,476$ "reducible" configurations and this used hundreds of hours of CPU time on a Cray computer.

A problem with Appel & Haken's proof is that the program ran for so long that it is impossible to verify manually. We cannot even to be sure that the hardware performed well enough, over such an extended period, to give a reliable result.

1996 Robertson, Sanders, Seymour and Thomas created a new proof, similar to Appel and Haken's, but more efficient: it requires checking only 633 reducible configurations. This must still be done by computer.

1976 Appel & Haken at the University of Illinois proved the 4–colour conjecture. Their proof required checking $1,476$ "reducible" configurations and this used hundreds of hours of CPU time on a Cray computer.

A problem with Appel & Haken's proof is that the program ran for so long that it is impossible to verify manually. We cannot even to be sure that the hardware performed well enough, over such an extended period, to give a reliable result.

1996 Robertson, Sanders, Seymour and Thomas created a new proof, similar to Appel and Haken's, but more efficient: it requires checking only 633 reducible configurations. This must still be done by computer.

2004 Werner and Gonthier gave an alternative proof using automatic theorem proving techniques. Again this requires us to trust a computer.

The 5 and 6-colour Theorems

Theorem 2.75.
Every simple planar graph G *is* 6*–colourable.*

The 5 and 6-colour Theorems

Theorem 2.75.
Every simple planar graph G *is* 6*–colourable.*

A proof of a 5–colour theorem can be found in most introductory texts on graph theory.

The 5 and 6-colour Theorems

Theorem 2.75.
Every simple planar graph G is 6–colourable.

A proof of a 5–colour theorem can be found in most introductory texts on graph theory.

In 1880 Tait made the following connection between 4–colouring of faces and edge–colouring.

Theorem 2.76. *Let G be a plane drawing of a graph which is connected, regular of degree three and has no bridges or loops. Then the faces of G can be coloured using 4 colours if and only if G has a proper edge–colouring using 3 colours.*