Polygons

Adam Klepáč

October 2, 2023

Contents

General Polygons

General Polygons – Definition

Polygon

A **polygon** is a closed 2D shape made of only segments.

General Polygons – Definition

Polygon

A **polygon** is a closed 2D shape made of only segments.

The endpoints of those segments are called **vertices**.

General Polygons – Definition

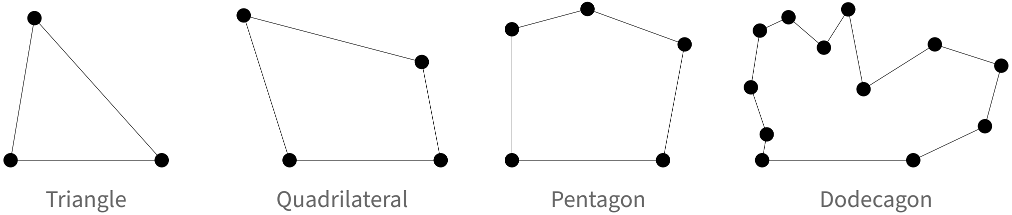
Polygon

A **polygon** is a closed 2D shape made of only segments.

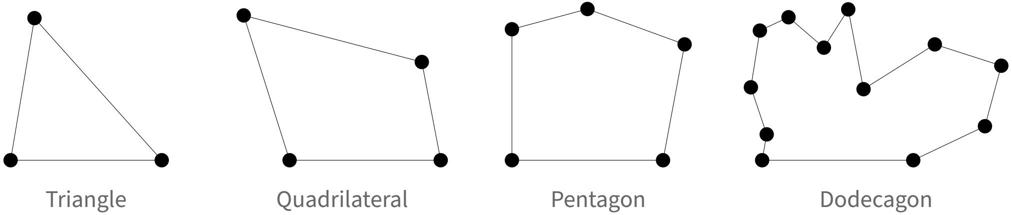
The endpoints of those segments are called **vertices**.

The segments themselves are called **edges**.

General Polygons – Examples

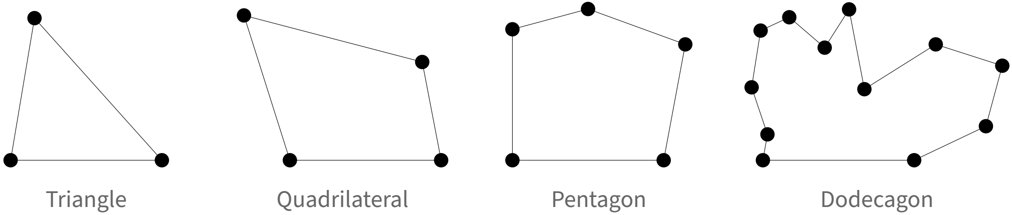


General Polygons – Examples



A polygon with $n \in \mathbb{N}$ sides is called an n-gon.

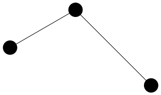
General Polygons – Examples



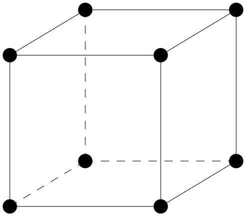
A polygon with $n \in \mathbb{N}$ sides is called an n-gon.

For example a polygon with 123456 sides is called a 123456-gon or decadismyriatrischilliatetrahectapentacontakaihexagon.

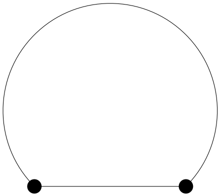
General Polygons – Counterexamples



Not closed



3D



Not straight

General Polygons – Convexity

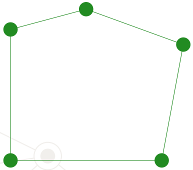
Convex Polygon

A polygon is called **convex** if it has no internal angle greater than 180°.

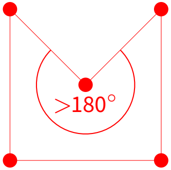
General Polygons – Convexity

Convex Polygon

A polygon is called **convex** if it has no internal angle greater than 180°.



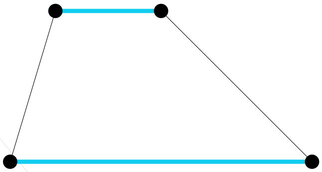
Convex



NOT convex

Convex Polygons

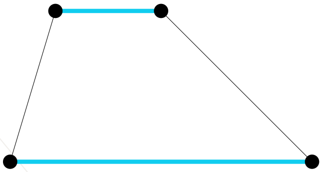
Convex Polygons – Special Types



Trapezoid/Trapezium

A convex quadrilateral with at least two parallel sides.

CONVEX POLYGONS – SPECIAL TYPES



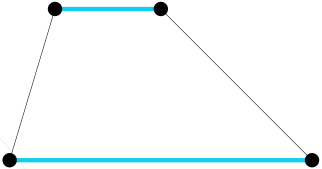
Trapezoid/Trapezium

A convex quadrilateral with at least two parallel sides.

Parallelogram

A convex quadrilateral with two pairs of parallel sides.

Convex Polygons – Special Types

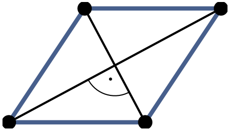


Trapezoid/Trapezium

A convex quadrilateral with at least two parallel sides.

Parallelogram

A convex quadrilateral with two pairs of parallel sides.



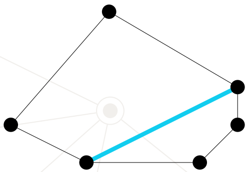
Rhombus

An **equilateral** (all sides of the same length) parallelogram.

Convex Polygons – Diagonals

Diagonal in a Convex Polygon

A diagonal of a **convex** polygon is a segment connecting two of its non-adjacent vertices.

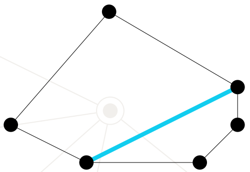


Diagonal in a convex hexagon.

Convex Polygons – Diagonals

Diagonal in a Convex Polygon

A **diagonal** of a **convex** polygon is a segment connecting two of its non-adjacent vertices.



Diagonal in a convex hexagon.

Voluntary HW: How many different diagonals does a convex n-gon have?

Convex Polygons – Triangulations

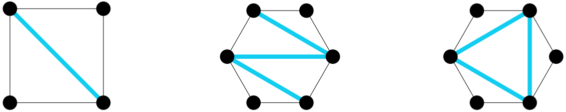
Triangulation of a Convex Polygon

A triangulation of a **convex** polygon is its division into triangles by non-intersecting diagonals.

Convex Polygons – Triangulations

Triangulation of a Convex Polygon

A triangulation of a **convex** polygon is its division into triangles by non-intersecting diagonals.

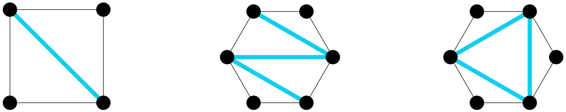


Examples of triangulations.

Convex Polygons – Triangulations

Triangulation of a Convex Polygon

A triangulation of a **convex** polygon is its division into triangles by non-intersecting diagonals.



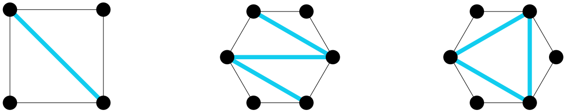
Examples of triangulations.

Voluntary HW: How many different triangulations of an n-gon are there?

Convex Polygons – Triangulations

Triangulation of a Convex Polygon

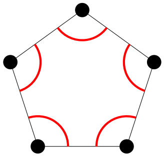
A triangulation of a **convex** polygon is its division into triangles by non-intersecting diagonals.



Examples of triangulations.

Voluntary HW: Find a **non-convex** polygon which **cannot** be triangulated.

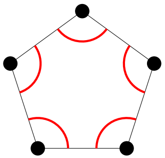
Convex Polygons – Internal Angles



Internal angles of a pentagon.

Question: What is the sum of internal angles of a convex n-gon?

Convex Polygons – Internal Angles

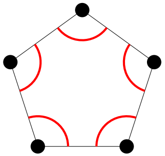


Internal angles of a pentagon.

Question: What is the sum of internal angles of a convex n-gon?

- For a triangle, it's 180°.

Convex Polygons – Internal Angles

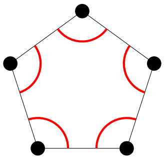


Internal angles of a pentagon.

Question: What is the sum of internal angles of a convex n-gon?

- For a triangle, it's 180°.
- For a square, it's 360°.

Convex Polygons – Internal Angles



Internal angles of a pentagon.

Question: What is the sum of internal angles of a convex n-gon?

- For a triangle, it's 180°.
- For a square, it's 360°.
- For a pentagon, it's 540°.

Convex Polygons – Internal Angles

We can count internal angles using triangulations.

Convex Polygons – Internal Angles

We can count internal angles using triangulations.

Into how many triangles is a convex n-gon divided?

Convex Polygons – Internal Angles

We can count internal angles using triangulations.

Into how many triangles is a convex *n*-gon divided?

Each triangle shares two vertices with an adjacent one.

Convex Polygons – Internal Angles

We can count internal angles using triangulations.

Into how many triangles is a convex *n*-gon divided?

Each triangle shares two vertices with an adjacent one.

We choose the first triangle – it covers 3 vertices.

Convex Polygons – Internal Angles

We can count internal angles using triangulations.

Into how many triangles is a convex *n*-gon divided?

Each triangle shares two vertices with an adjacent one.

We choose the first triangle – it covers 3 vertices.

After that, each triangle covers only one more vertex.

Convex Polygons – Internal Angles

We can count internal angles using triangulations.

Into how many triangles is a convex n-gon divided?

Each triangle shares two vertices with an adjacent one.

We choose the first triangle – it covers 3 vertices.

After that, each triangle covers only one more vertex.

This means, that an n-gon is divided into $n-2$ triangles.

Convex Polygons – Internal Angles

We can count internal angles using triangulations.

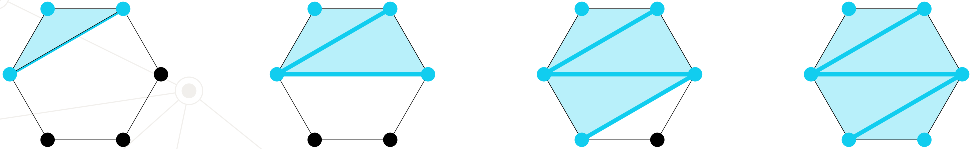
Into how many triangles is a convex n-gon divided?

Each triangle shares two vertices with an adjacent one.

We choose the first triangle – it covers 3 vertices.

After that, each triangle covers only one more vertex.

This means, that an n-gon is divided into $n - 2$ triangles.



Construction of a triangulation of a hexagon.

Convex Polygons – Internal Angles

A convex n-gon is divided into $n - 2$ triangles.

Convex Polygons – Internal Angles

A convex n-gon is divided into $n-2$ triangles.

The sum of all internal angles in a triangle is 180°.

Convex Polygons – Internal Angles

A convex n-gon is divided into $n-2$ triangles.

The sum of all internal angles in a triangle is 180°.

Sum of Internal Angles in a Convex Polygon

The sum of all internal angles of a convex n-gon is $(n-2)\cdot 180^\circ$.

Regular Polygons

Definition

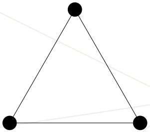
Regular Polygon

A **regular polygon** is a convex polygon whose sides all have the same length and whose internal angles all have the same size.

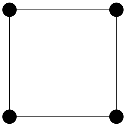
Definition

Regular Polygon

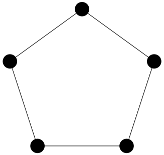
A **regular polygon** is a convex polygon whose sides all have the same length and whose internal angles all have the same size.



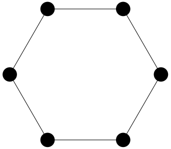
Equilateral triangle
(regular trigon)



Square (regular tetragon)



Regular pentagon



Regular hexagon

Review – Plane Transformations

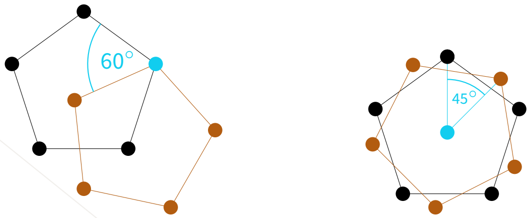
Rotation

Rotation of a polygon consists of well ... rotating each of its points by a fixed angle around a fixed point (called *anchor*).

Review – Plane Transformations

Rotation

Rotation of a polygon consists of well ... rotating each of its points by a fixed angle around a fixed point (called *anchor*).



Examples of rotations.

Review – Plane Transformations

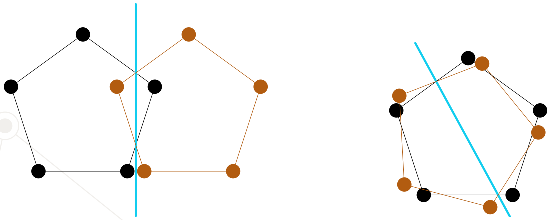
Reflection

Reflection of a polygon consists of ‘mirroring’ each of its points through a given line (called *axis of reflection*).

Review – Plane Transformations

Reflection

Reflection of a polygon consists of 'mirroring' each of its points through a given line (called *axis of reflection*).



Examples of reflections.

Review – Plane Transformations

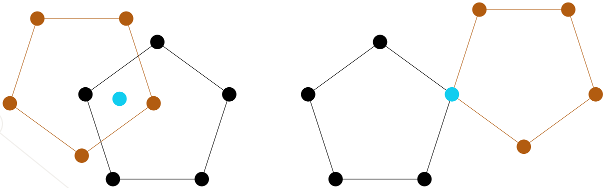
Point Symmetry

Point symmetry of a polygon consists of ‘mirroring’ each of its points through a given point (called *center of symmetry*).

Review – Plane Transformations

Point Symmetry

Point symmetry of a polygon consists of 'mirroring' each of its points through a given point (called *center of symmetry*).



Examples of point symmetries.

Symmetries of Regular Polygons

Question: What are the transformations that don't change regular polygons in any way?

Symmetries of Regular Polygons

Question: What are the transformations that don't change regular polygons in any way?

- rotational symmetries
  - rotation by $\frac{k \cdot 360°}{n}$ where k is any number between 1 and n

Symmetries of Regular Polygons

Question: What are the transformations that don't change regular polygons in any way?

- rotational symmetries
  - rotation by $\frac{k \cdot 360^\circ}{n}$ where k is any number between 1 and n
- reflectional (line) symmetries

Symmetries of Regular Polygons

Question: What are the transformations that don't change regular polygons in any way?

- rotational symmetries
  - rotation by $\frac{k \cdot 360^\circ}{n}$ where k is any number between 1 and n
- reflectional (line) symmetries
  - for n even reflections over lines passing through centres of opposite sides
  - for n even over lines passing through opposite vertices
  - for n odd over lines passing through a centre of a side and the opposite vertex

Symmetries of Regular Polygons

Question: What are the transformations that don't change regular polygons in any way?

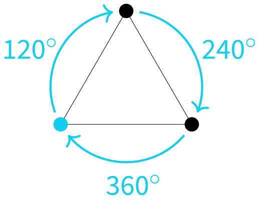
- rotational symmetries
  - rotation by $\frac{k \cdot 360^\circ}{n}$ where k is any number between 1 and n
- reflectional (line) symmetries
  - for n even reflections over lines passing through centres of opposite sides
  - for n even over lines passing through opposite vertices
  - for n odd over lines passing through a centre of a side and the opposite vertex
- point symmetries

Symmetries of Regular Polygons

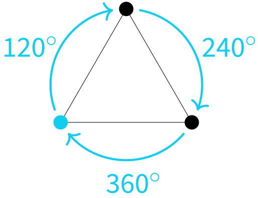
Question: What are the transformations that don't change regular polygons in any way?

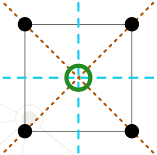
- rotational symmetries
  - rotation by $\frac{k \cdot 360^\circ}{n}$ where k is any number between 1 and n
- reflectional (line) symmetries
  - for n even reflections over lines passing through centres of opposite sides
  - for n even over lines passing through opposite vertices
  - for n odd over lines passing through a centre of a side and the opposite vertex
- point symmetries
  - only through the 'centre' – the point where its axes of symmetry intersect – in case n is even

Symmetries of Regular Polygons

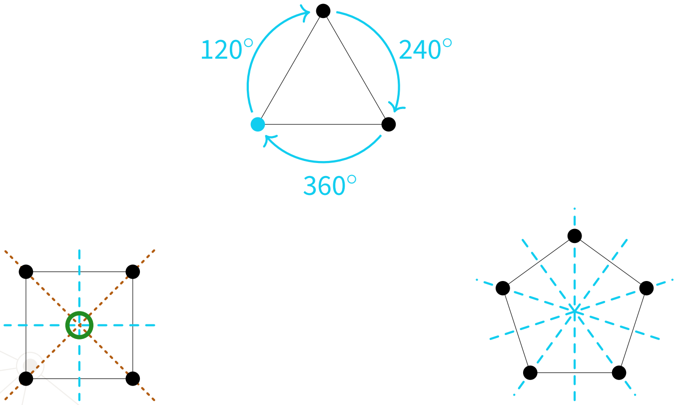


Symmetries of Regular Polygons





Symmetries of Regular Polygons



Examples of regular polygon symmetries

Cryptography on Regular Polygons

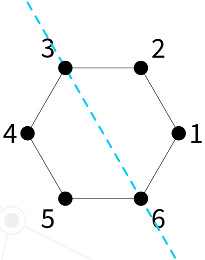
Chaining Symmetries

Given two symmetries, s_1 and s_2 of a regular polygon, one can apply them one after the other (‘compose’ them, like functions).

Chaining Symmetries

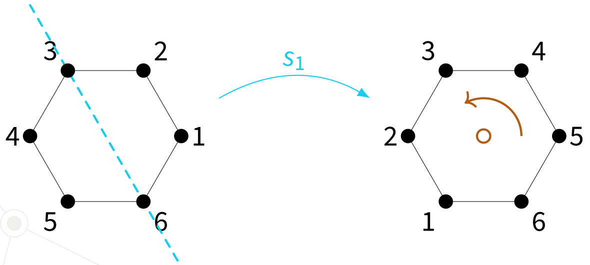
Given two symmetries, s_1 and s_2 of a regular polygon, one can apply them one after the other ('compose' them, like functions).
We'll denote this composition simply by $s_1 s_2$.

Chaining Symmetries – Example



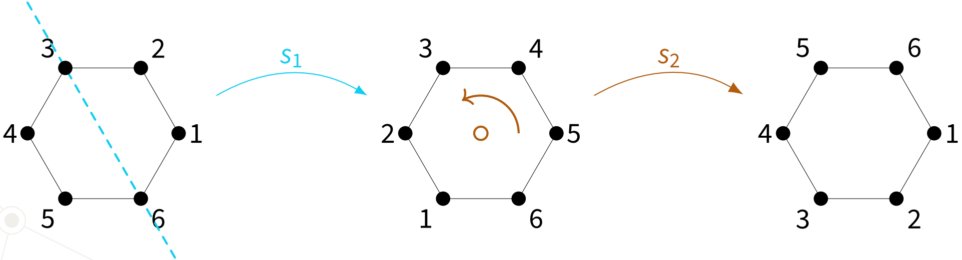
Example of a chain of symmetries.

Chaining Symmetries – Example



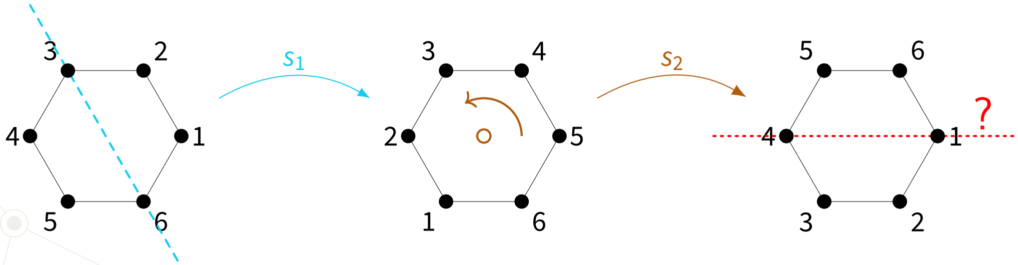
Example of a chain of symmetries.

Chaining Symmetries – Example



Example of a chain of symmetries.

Chaining Symmetries – Example



Example of a chain of symmetries.

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Two symmetries can ‘combine’ to create a different symmetry.

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Two symmetries can ‘combine’ to create a different symmetry.

Natural question: How many (and which) symmetries of a regular polygon do I need to get all the others?

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Two symmetries can 'combine' to create a different symmetry.

Natural question: How many (and which) symmetries of a regular polygon do I need to get all the others?

For example,

- if s_1 is any reflectional symmetry and s_2 is a rotation by 60° counter-clockwise, then $s_2^3 s_1$ (s_2^3 means $s_2 s_2 s_2$) reflects a hexagon through a line perpendicular to the line of s_1.

Chaining Symmetries – How Many Do We Need?

Discounting point symmetry, an n-gon has $2n$ symmetries.

Two symmetries can 'combine' to create a different symmetry.

Natural question: How many (and which) symmetries of a regular polygon do I need to get all the others?

For example,

- if s_1 is any reflectional symmetry and s_2 is a rotation by 60° counter-clockwise, then $s_2^3 s_1$ (s_2^3 means $s_2 s_2 s_2$) reflects a hexagon through a line perpendicular to the line of s_1.
- if s_1 is a rotation by 120° clockwise and s_2 is a reflection through a vertical line passing through the top vertex, then $s_1 s_2$ is a reflection through the line given by the rotation of the line of s_2 60° clockwise.

Chaining Symmetries – How Many Do We Need?

Actually, for a general n-gon, we need only **two**:

Chaining Symmetries – How Many Do We Need?

Actually, for a general n-gon, we need only **two**:

- rotation by $360^\circ/n$ in any direction (we'll denote it r),

Chaining Symmetries – How Many Do We Need?

Actually, for a general n-gon, we need only **two**:

- rotation by $360^\circ/n$ in any direction (we'll denote it r),
- any reflection (we'll denote it s).

Chaining Symmetries – Triangle

Let r be the rotation by 120° and s any reflectional symmetry.

Chaining Symmetries – Triangle

Let r be the rotation by 120° and s any reflectional symmetry.

- The other two rotational symmetries are r^2 and r^3.

Chaining Symmetries – Triangle

Let r be the rotation by 120° and s any reflectional symmetry.

- The other two rotational symmetries are r^2 and r^3.
- The other two reflectional symmetries are rs and r^2s.

Chaining Symmetries – Triangle

Let r be the rotation by 120° and s any reflectional symmetry.

- The other two rotational symmetries are r^2 and r^3.
- The other two reflectional symmetries are rs and r^2s.
- Therefore, all the symmetries of an equilateral triangle are

$$\{r, r^2, r^3, s, rs, r^2s\}.$$

Chaining Symmetries – General Algorithm

In general, to create all symmetries, one needs a rotation by an angle $k \cdot 360^\circ / n$ where k **doesn't share a prime factor** with n (in other words, the fraction $\frac{k}{n}$ cannot be simplified) and any one reflectional symmetry.

Chaining Symmetries – General Algorithm

In general, to create all symmetries, one needs a rotation by an angle $k \cdot 360^\circ / n$ where k **doesn't share a prime factor** with n (in other words, the fraction $\frac{k}{n}$ cannot be simplified) and any one reflectional symmetry.

Why?

- If k shares factors with n, then you can never get rotation by $360^\circ / n$.

Chaining Symmetries – General Algorithm

In general, to create all symmetries, one needs a rotation by an angle $k \cdot 360^\circ/n$ where k **doesn't share a prime factor** with n (in other words, the fraction $\frac{k}{n}$ cannot be simplified) and any one reflectional symmetry.

Why?

- If k shares factors with n, then you can never get rotation by $360^\circ/n$.
- Two symmetries cannot in general produce every rotation.

Chaining Symmetries – General Algorithm

In general, to create all symmetries, one needs a rotation by an angle $k \cdot 360^\circ/n$ where k **doesn't share a prime factor** with n (in other words, the fraction $\frac{k}{n}$ cannot be simplified) and any one reflectional symmetry.

Why?

- If k shares factors with n, then you can never get rotation by $360^\circ/n$.
- Two symmetries cannot in general produce every rotation.
- Two rotations can never produce a symmetry.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ / n$ such that k doesn't share factors with n and a reflectional symmetry s.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ / n$ such that k doesn't share factors with n and a reflectional symmetry s.

If you need to calculate a rotation, then

1. First measure the angle **counter-clockwise**.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ / n$ such that k doesn't share factors with n and a reflectional symmetry s.

If you need to calculate a rotation, then

1. First measure the angle **counter-clockwise**.
2. Find a such that r^a is the rotation by $360^\circ / n$.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ / n$ such that k doesn't share factors with n and a reflectional symmetry s.

If you need to calculate a rotation, then

1. First measure the angle **counter-clockwise**.
2. Find a such that r^a is the rotation by $360^\circ / n$.
3. Then, find b such that $(r^a)^b = r^{ab}$ is your desired rotation.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ / n$ such that k doesn't share factors with n and a line symmetry s.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360° / n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360° / n$.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360^\circ/n$.
2. Determine the angle **in any direction** between the lines of your given reflection s and the reflection you want.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360^\circ/n$.
2. Determine the angle **in any direction** between the lines of your given reflection s and the reflection you want.
3. Find b such that r^{ab} is a rotation **in the opposite direction** by **twice** the angle from the previous step.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360^\circ/n$ such that k doesn't share factors with n and a line symmetry s.

If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360^\circ/n$.
2. Determine the angle **in any direction** between the lines of your given reflection s and the reflection you want.
3. Find b such that r^{ab} is a rotation **in the opposite direction** by **twice** the angle from the previous step.
4. $r^{ab}s$ is your desired reflection.

Chaining Symmetries – General Algorithm

You're given a rotation r by $k \cdot 360° / n$ such that k doesn't share factors with n and a line symmetry s.

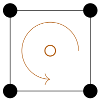
If you need to calculate a reflection, then

1. Find a such that r^a is the rotation by $360° / n$.
2. Determine the angle **in any direction** between the lines of your given reflection s and the reflection you want.
3. Find b such that r^{ab} is a rotation **in the opposite direction** by **twice** the angle from the previous step.
4. $r^{ab}s$ is your desired reflection.

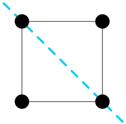
Voluntary HW: *Why* does this algorithm work?

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square:

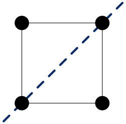


rotation r by 270°



reflection s over the first diagonal

and want to produce



reflection over the second diagonal

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square: rotation r by 270° counter-clockwise and reflection s over the first diagonal.

How to produce reflection over the other diagonal?

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square: rotation r by 270° counter-clockwise and reflection s over the first diagonal.
How to produce reflection over the other diagonal?

We use the algorithm.

1. Repeating r three times gives the rotation by 90° counter-clockwise, that is, $a = 3$.

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square: rotation r by 270° counter-clockwise and reflection s over the first diagonal.
How to produce reflection over the other diagonal?

We use the algorithm.

1. Repeating r three times gives the rotation by 90° counter-clockwise, that is, $a = 3$.
2. The angle between the two diagonals is 90° in any direction.

Chaining Symmetries – Algorithm Example

We're given two symmetries of the square: rotation r by 270° counter-clockwise and reflection s over the first diagonal.

How to produce reflection over the other diagonal?

We use the algorithm.

1. Repeating r three times gives the rotation by 90° counter-clockwise, that is, $a = 3$.
2. The angle between the two diagonals is 90° in any direction.
3. Repeating the rotation from step 1 two times (that is, $b = 2$) and then using s gives the desired symmetry – in this case it's $(r^3)^2 s = r^6 s$. Of course, r^4 is rotation by 360° which does nothing, so the final symmetry is $r^2 s$.

1

Encoding Messages Using Symmetries

Encoding Messages – General Idea

- Send a message as a sequence of strings which **can be decoded on the other side**.

Encoding Messages – General Idea

- Send a message as a sequence of strings which **can be decoded on the other side**.
- The decoding should require a ‘key’ which is **statistically impossible to determine quickly**.

Encoding Messages – General Idea

- Send a message as a sequence of strings which **can be decoded on the other side**.
- The decoding should require a ‘key’ which is **statistically impossible to determine quickly**.
- The encoding and decoding must be done procedurally – requires a system with concrete rules and a **limited** (but huge) **number of combinations**.

Encoding Messages – Regular Polygons

How can one use regular polygons to encode messages?

Encoding Messages – Regular Polygons

How can one use regular polygons to encode messages?

Quite easily, assign symbols to vertices and choose one **main** vertex.

Encoding Messages – Regular Polygons

How can one use regular polygons to encode messages?

Quite easily, assign symbols to vertices and choose one **main** vertex.

The symbol in the **main** vertex is what is sent during each tick (time interval during which symbols are sent).

Encoding Messages – Regular Polygons

How can one use regular polygons to encode messages?

Quite easily, assign symbols to vertices and choose one **main** vertex.

The symbol in the **main** vertex is what is sent during each tick (time interval during which symbols are sent).

Rotations and reflections ‘move symbols between vertices’.

Encoding Messages – Regular Polygons

How can one use regular polygons to encode messages?

Quite easily, assign symbols to vertices and choose one **main** vertex.

The symbol in the **main** vertex is what is sent during each tick (time interval during which symbols are sent).

Rotations and reflections ‘move symbols between vertices’.

This means that after applying a rotation or reflection, a (in most cases) different symbol will appear in the main vertex.

Encoding Messages – Morse Code

The Morse Code has two symbols: $-$ and $\cdot$.

Encoding Messages – Morse Code

The Morse Code has two symbols: — and ·.

We also need to be able to determine spaces between words – we can assign one vertex a 'blank', meaning space.

Encoding Messages – Morse Code

The Morse Code has two symbols: — and ·.

We also need to be able to determine spaces between words – we can assign one vertex a 'blank', meaning space.

This means that to send messages written Morse Code using regular polygons, we need three vertices – a triangle.

Encoding Messages – Morse Code

Let the top vertex be main and let's choose a rotation $r = \circlearrowleft 120^\circ$ and a reflection s depicted below.

Encoding Messages – Morse Code

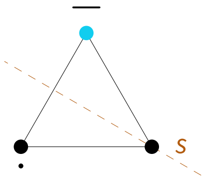
Let the top vertex be main and let's choose a rotation $r = \circlearrowleft 120^\circ$ and a reflection s depicted below.

We should also assign Morse Code symbols to two of the vertices. We can do so randomly.

Encoding Messages – Morse Code

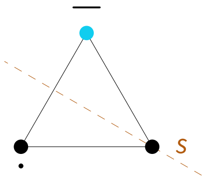
Let the top vertex be main and let's choose a rotation $r = \circlearrowleft 120°$ and a reflection s depicted below.

We should also assign Morse Code symbols to two of the vertices. We can do so randomly.



Encoding Messages – Morse Code

Let the top vertex be main and let's choose a rotation $r = \circlearrowleft 120^\circ$ and a reflection s depicted below.
We should also assign Morse Code symbols to two of the vertices. We can do so randomly.



The top vertex is the main one, which means that the symbol above it will get sent.

Encoding Messages – Morse Code

The important property of symmetries is that two different sequences can give the same symmetry.

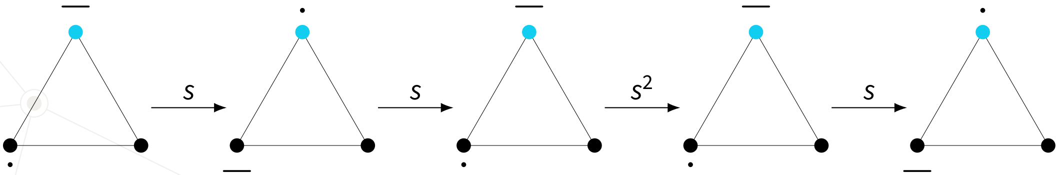
Encoding Messages – Morse Code

The important property of symmetries is that two different sequences can give the same symmetry. Meaning, we can encode one message **MANY** different ways.

Encoding Messages – Morse Code

The important property of symmetries is that two different sequences can give the same symmetry. Meaning, we can encode one message **MANY** different ways.

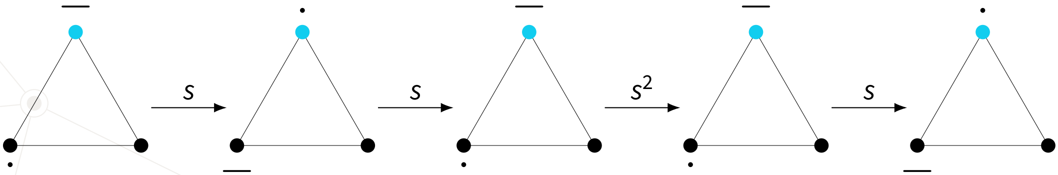
For example, we can send the letter P (· — — ·) like this:



Encoding Messages – Morse Code

The important property of symmetries is that two different sequences can give the same symmetry. Meaning, we can encode one message **MANY** different ways.

For example, we can send the letter P ($\cdot - - \cdot$) like this:



Or, as a sequence

$$s, s, s^2, s.$$

Encoding Messages – Morse Code

Notice that we didn't need the rotation *r* at all for this example.

Encoding Messages – Morse Code

Notice that we didn't need the rotation r at all for this example.

However, that doesn't mean we **can't** use it.

Encoding Messages – Morse Code

Notice that we didn't need the rotation r at all for this example.

However, that doesn't mean we **can't** use it.

Actually, it's often beneficial to encode messages using different combinations of symmetries for security.

Encoding Messages – Morse Code

Notice that we didn't need the rotation r at all for this example.

However, that doesn't mean we **can't** use it.

Actually, it's often beneficial to encode messages using different combinations of symmetries for security.

The letter $\cdot - - \cdot$ can also be sent for example like this

$$r^2, sr, sr^2, rsr.$$