Centralised Connectivity-Preserving Transformations for Programmable Matter: A Minimal Seed Approach

M. Connor, O. Michail, I. Potapov

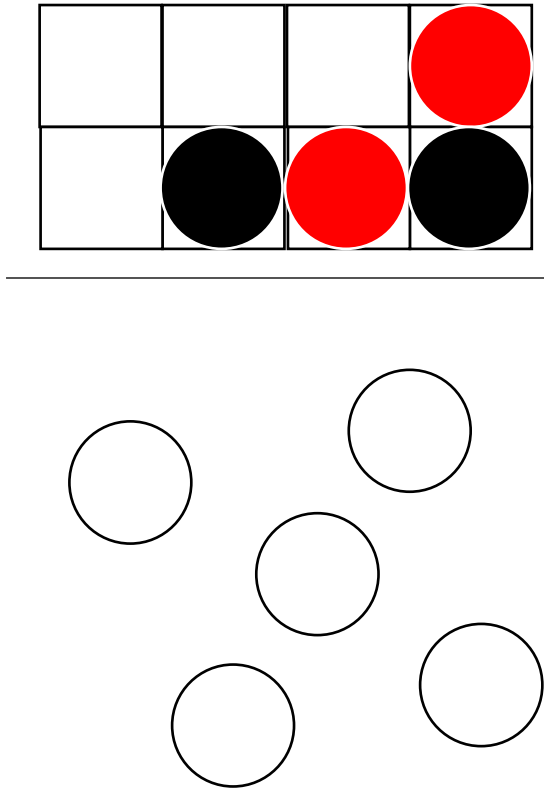
ALGOSENSORS 2021

Programmable matter systems

Multi-Agent systems

Decentralised

Weak



Overview

- Model and problem definition
- Blocked shapes
- Line to nice shape
- Nice shape to nice shape

Model and Problem definitions

Set S of n agents in the shape A occupying cells in a 2D grid

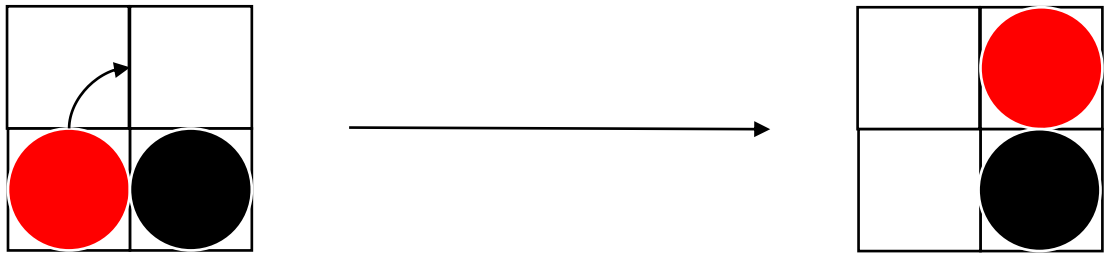
Transformation of A into shape B in time t by a series of configurations $C_0 \ldots C_t$ each reachable by a single move from a single node

Centralised model to explore feasibility

Rotation – movement of one node 90° around another node

Rot-Transformability – Rotation only

RotC-Transformability – Rotation only, connectivity must be preserved



Our focus – transforming lines into nice shapes in the RotC setting

Lines cannot meaningly transform in RotC

Therefore transformations are aided by **seeds** – nodes placed in empty cells neighbouring a shape to create a new shape to aid transformation

We discard nodes at the end – the seed and any **waste**

Related Work

Various programmable matter models developed e.g. [Dumitrescu, Pach, ACM, 2004]

Programmable materials developed e.g. [Rothemund, Nature, 2006]

Recent papers on the concept of seed-assisted transformations in programmable matter

- Universal transformation for RoT-Transformability given colour-consistency [Michail et al, JCSS, 2020]
- Universal transformation with connectivity preservation using “leapfrog” and “monkey” movement and a 5-node seed [Akitaya et al, Algorithmica, 2021]

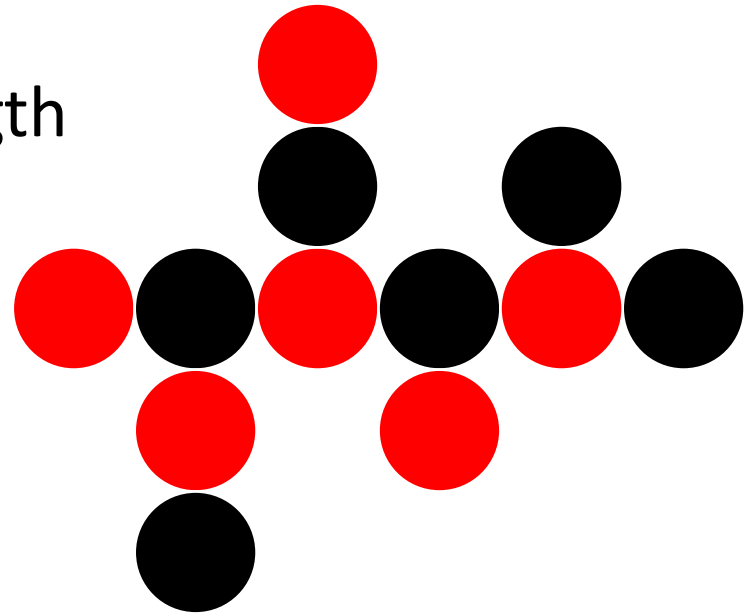
Nice shapes

Class defined in [Almethen et al, Theoretical Computer Science, 2020]

All nodes are part of a central line or lines perpendicular to it

We use two classes:

- M - The perpendicular lines must be of even length
- N – Any nice shape



Main Results

Theorem 3. A line of length n can be transformed to any given nice shape in the class M_{n_1} using a 3-node seed in $O(n^2)$ time.

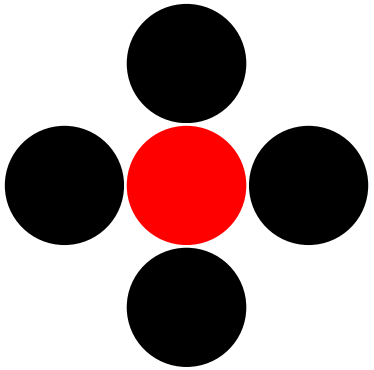
Theorem 4. A nice shape in the class M_n can be transformed to any given nice shape from M_n using a 4-node seed in $O(n^2)$ time.

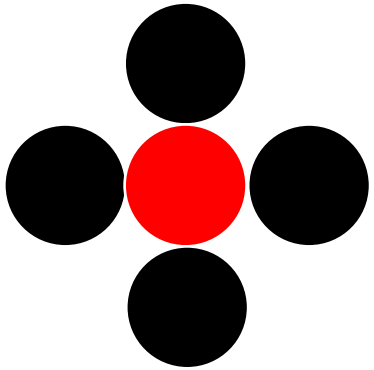
Theorem 6. A nice shape of n nodes can be transformed to any given nice shape N_n using a 4-node seed in $O(n^2)$ time.

Blocked shapes – Rotation only

If nodes on the exterior of a shape are not connected at the edge to other nodes on the exterior, then the shape is blocked under the conditions of RoT-Transformability.

- One node – trivially blocked
- Interior nodes – blocked by nodes around them
- Exterior nodes – blocked by other exterior nodes
- Therefore, all nodes are blocked

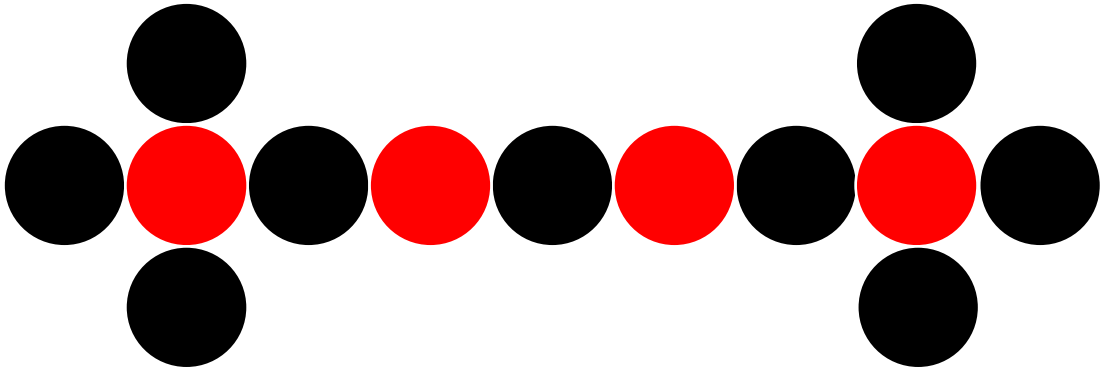




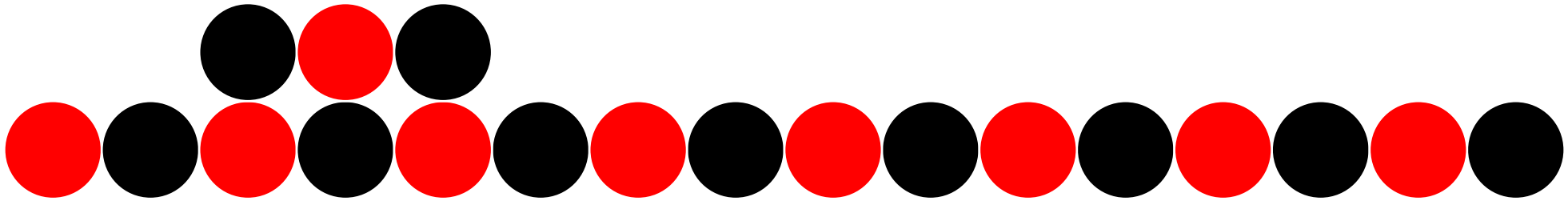
Blocked shapes - Connectivity

Shapes made up of lines connecting such shapes are blocked under the conditions of RoTC-Transformability.

- Rotation only shapes are blocked
- Lines – blocked because each is the only node connecting two shapes
- The shapes cannot enable movement of nodes in the lines

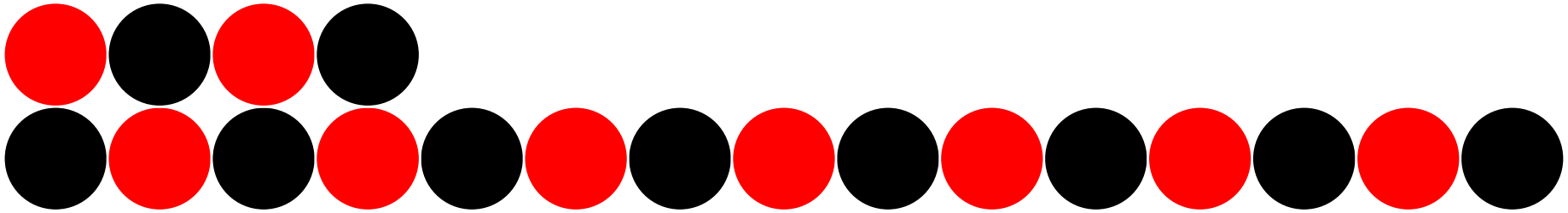


Line to nice shape



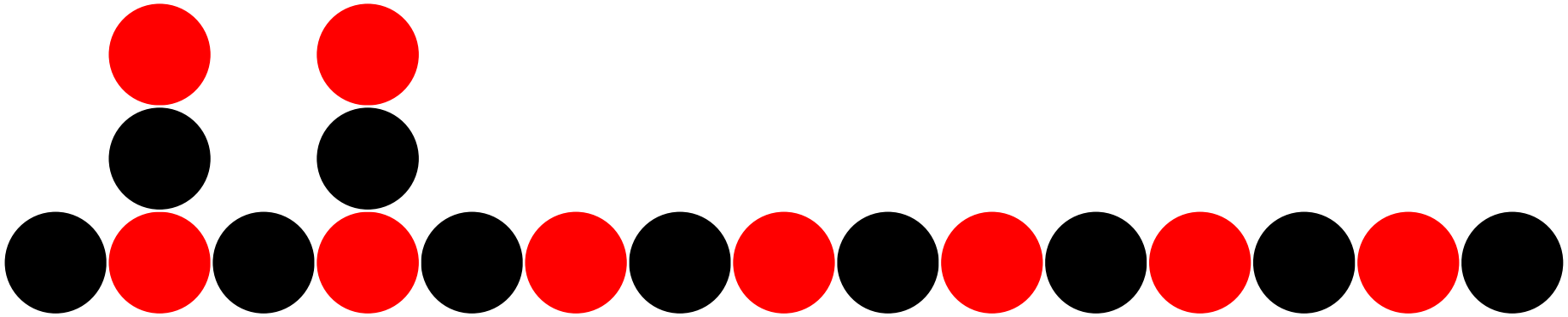
Line to nice shape

Step 1 – Raise nodes from the line using the seed



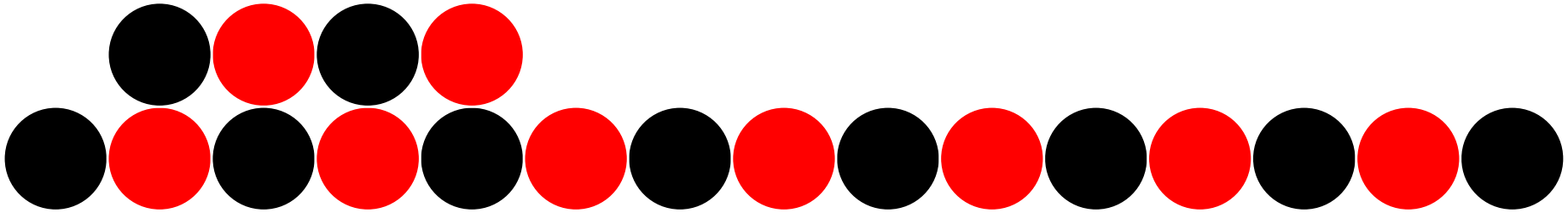
Line to nice shape

Step 1 – Raise nodes from the line using the seed



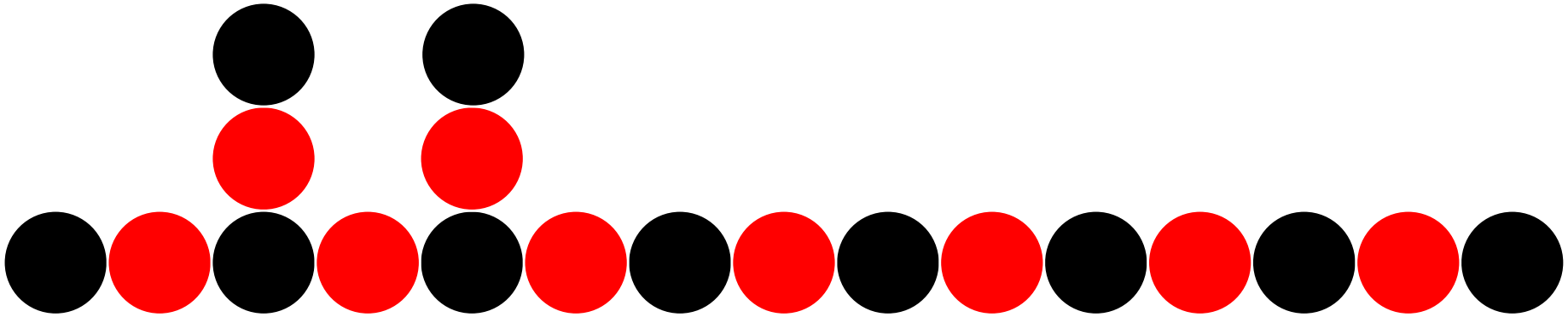
Line to nice shape

Step 1 – Raise nodes from the line using the seed



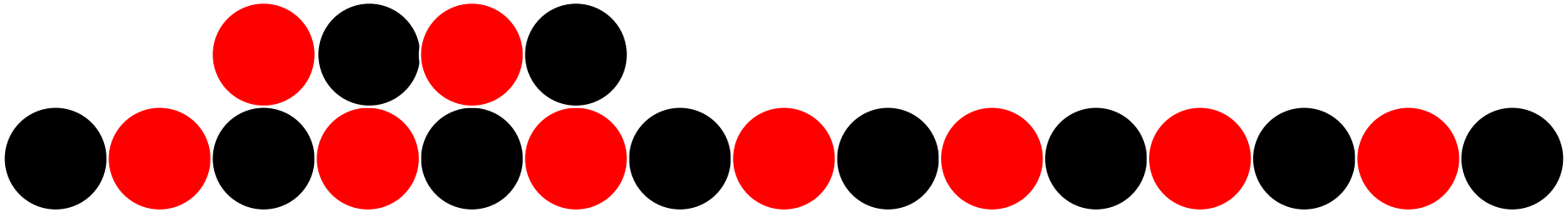
Line to nice shape

Step 1 – Raise nodes from the line using the seed



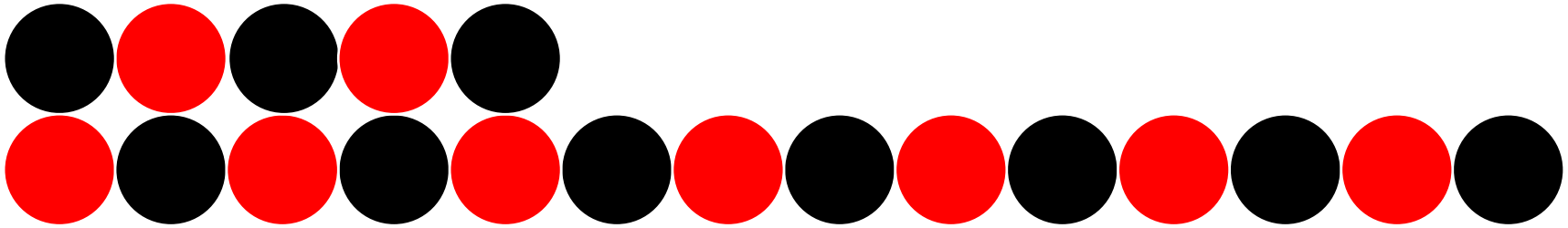
Line to nice shape

Step 1 – Raise nodes from the line using the seed



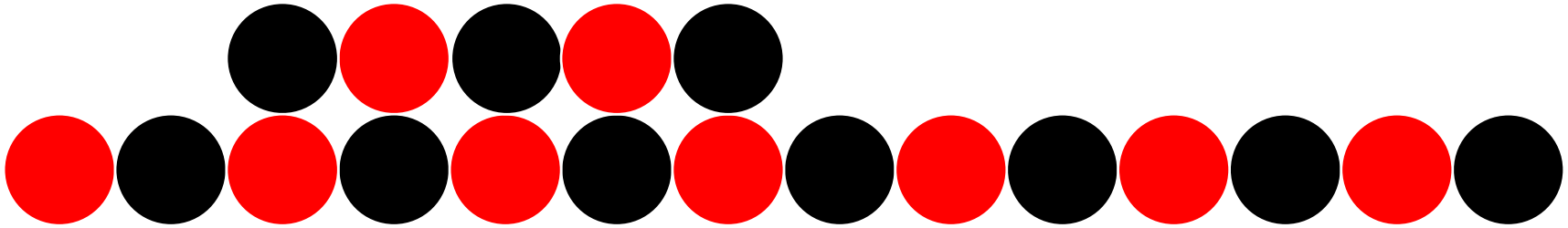
Line to nice shape

Step 1 – Raise nodes from the line using the seed



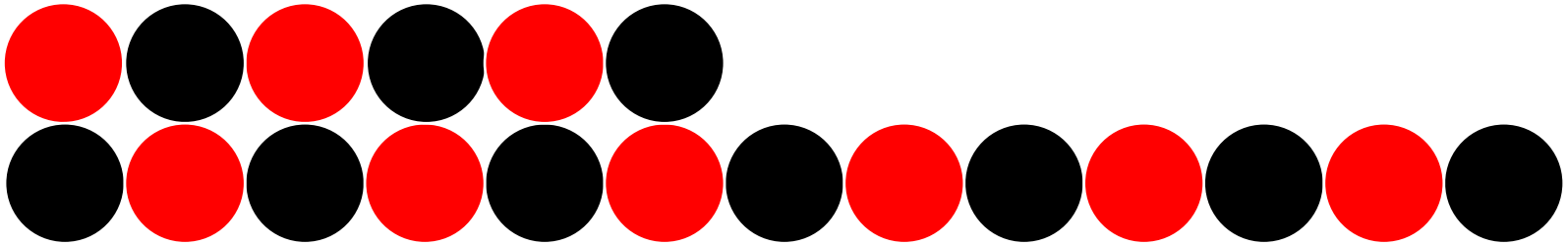
Line to nice shape

Step 1 – Raise nodes from the line using the seed



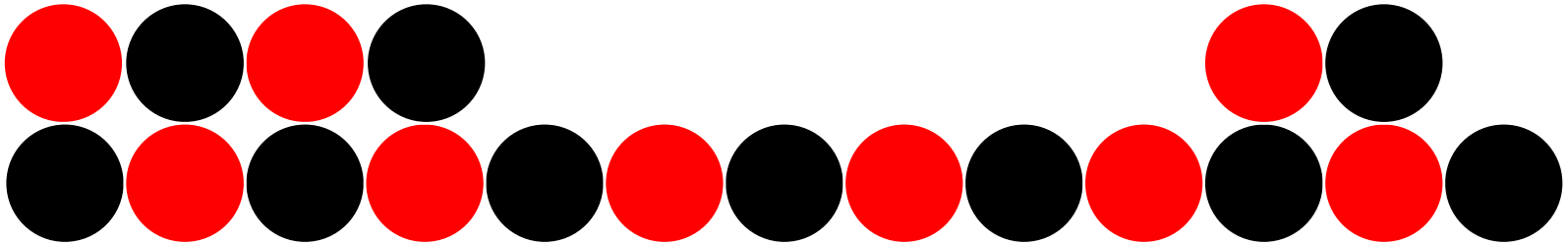
Line to nice shape

Step 1 – Raise nodes from the line using the seed



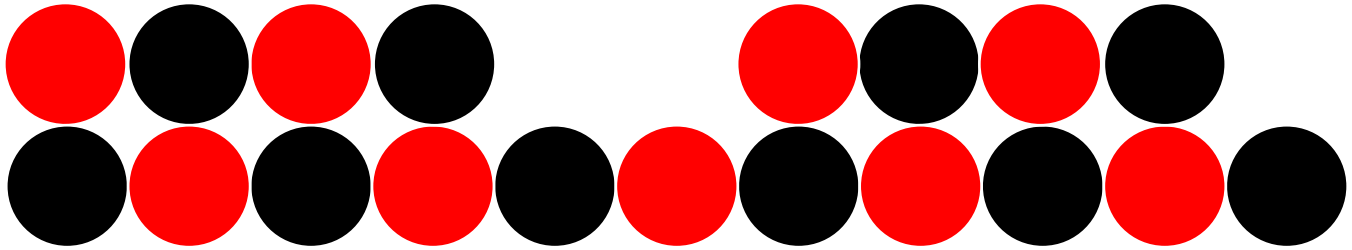
Line to nice shape

Step 1 – Raise nodes from the line using the seed



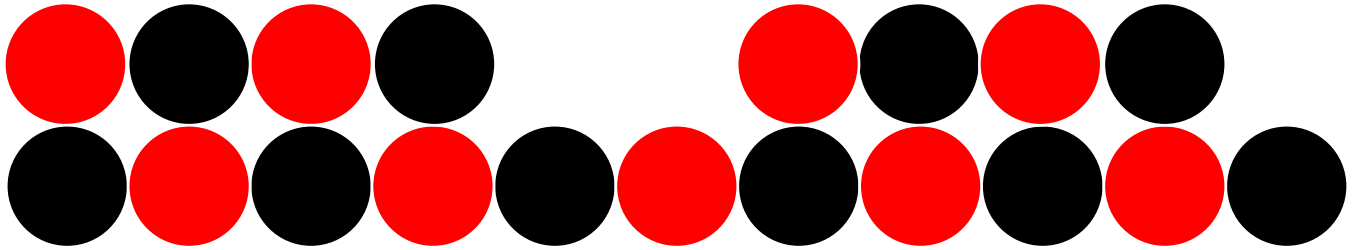
Line to nice shape

Step 1 – Raise nodes from the line using the seed



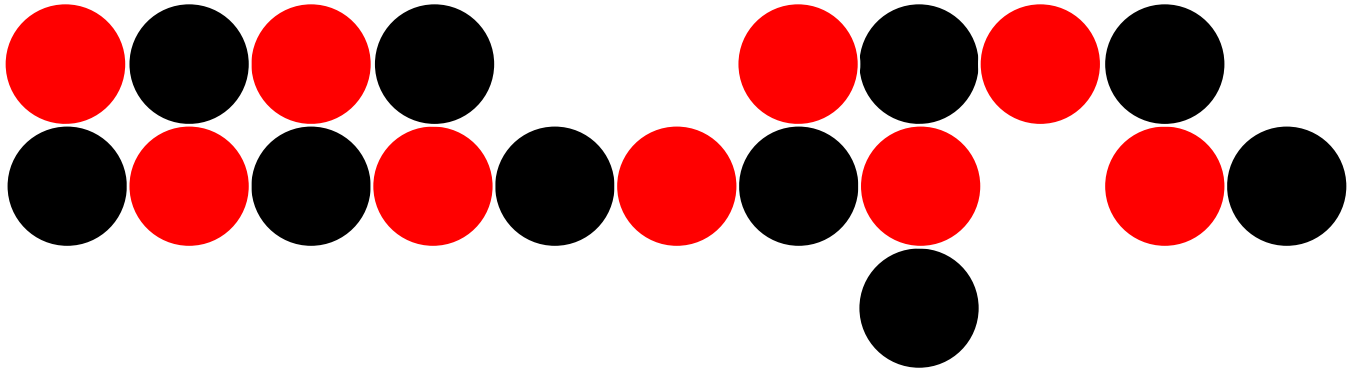
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



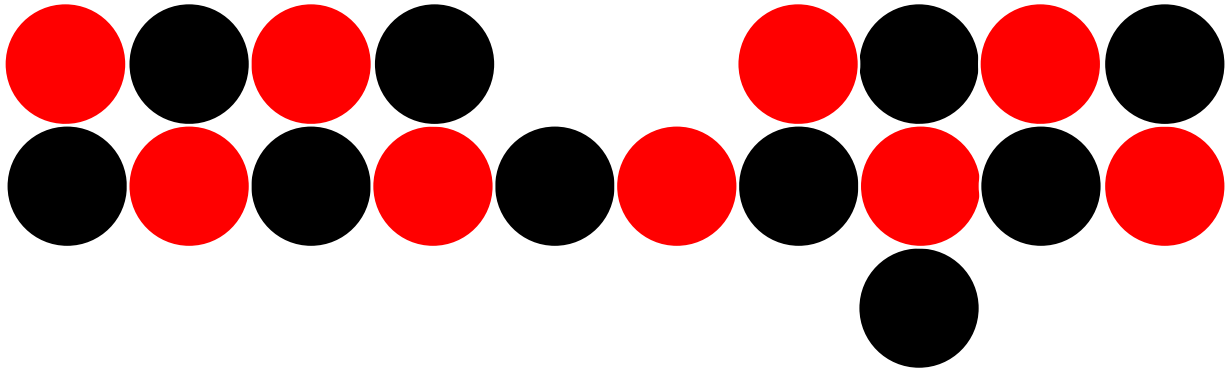
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



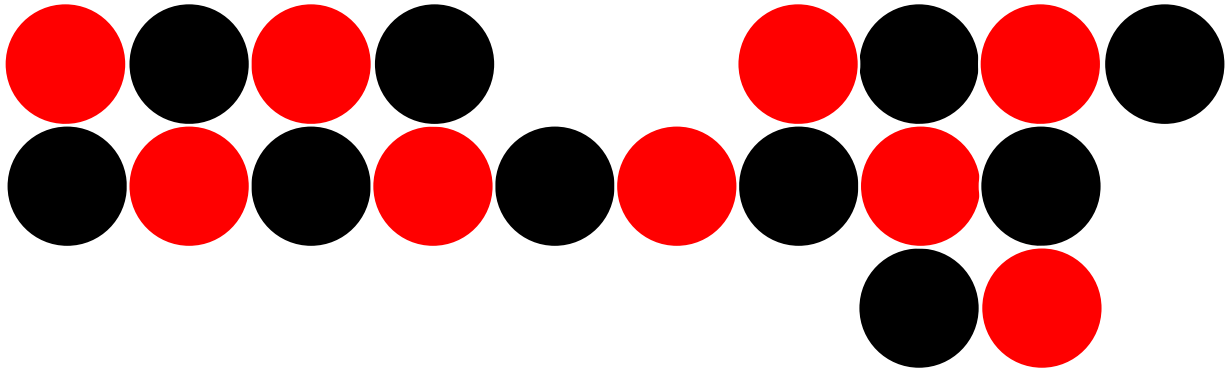
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



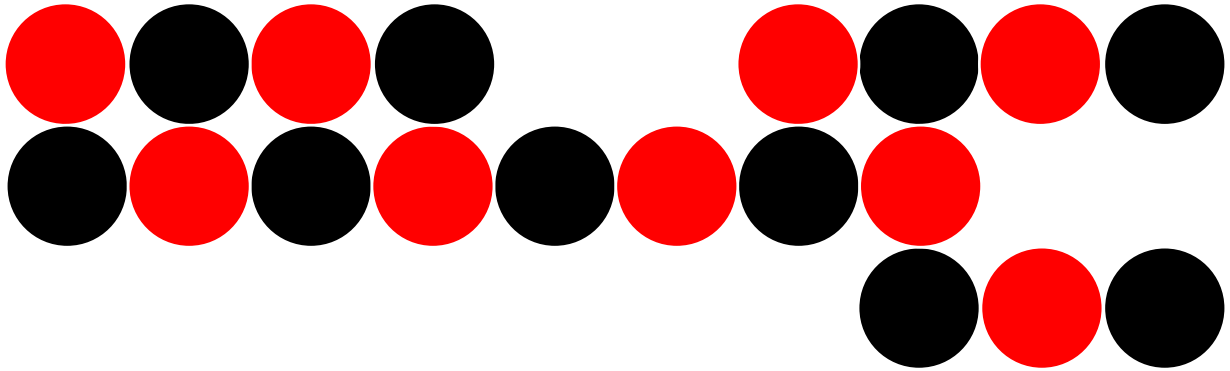
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



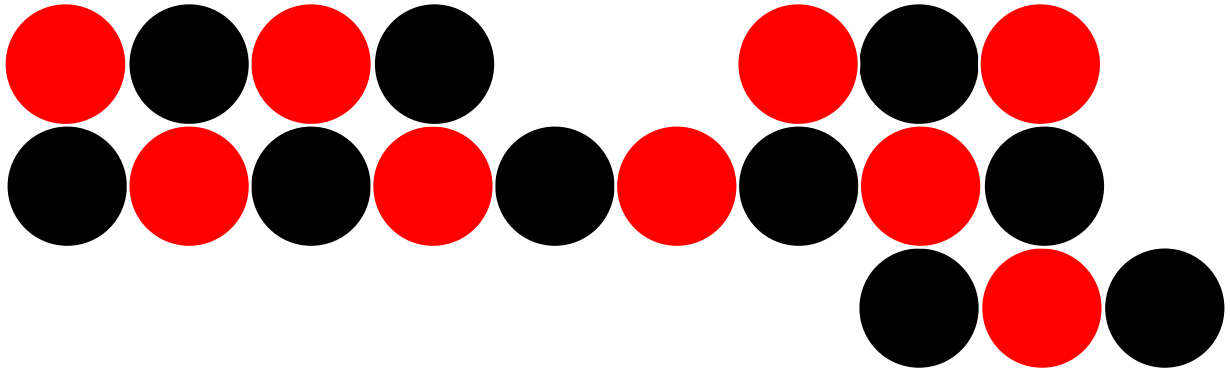
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



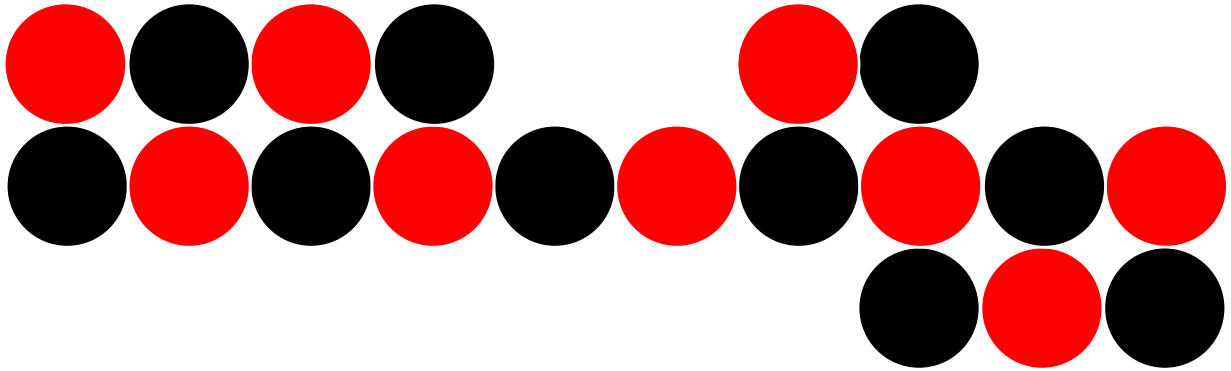
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



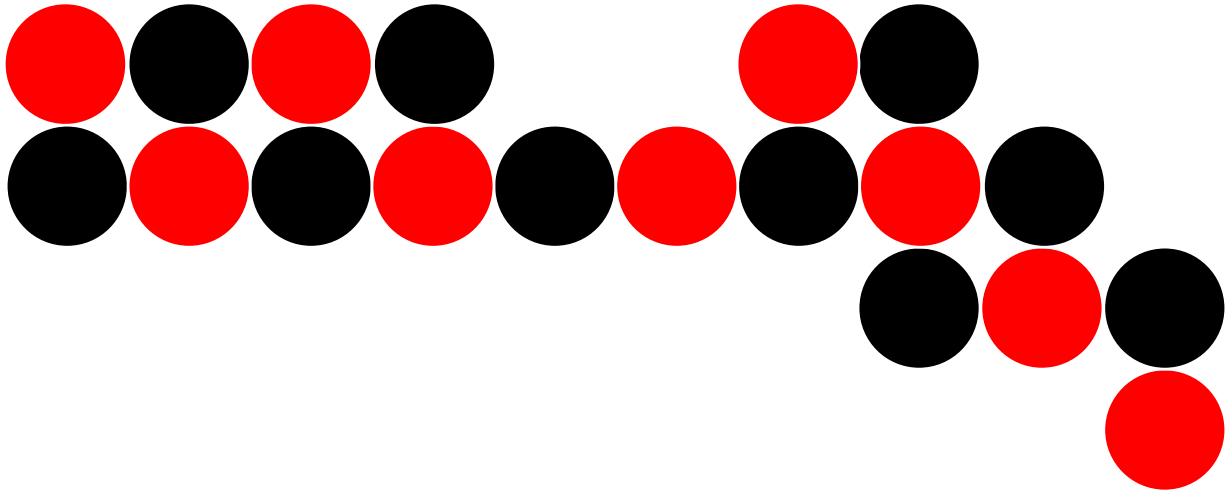
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



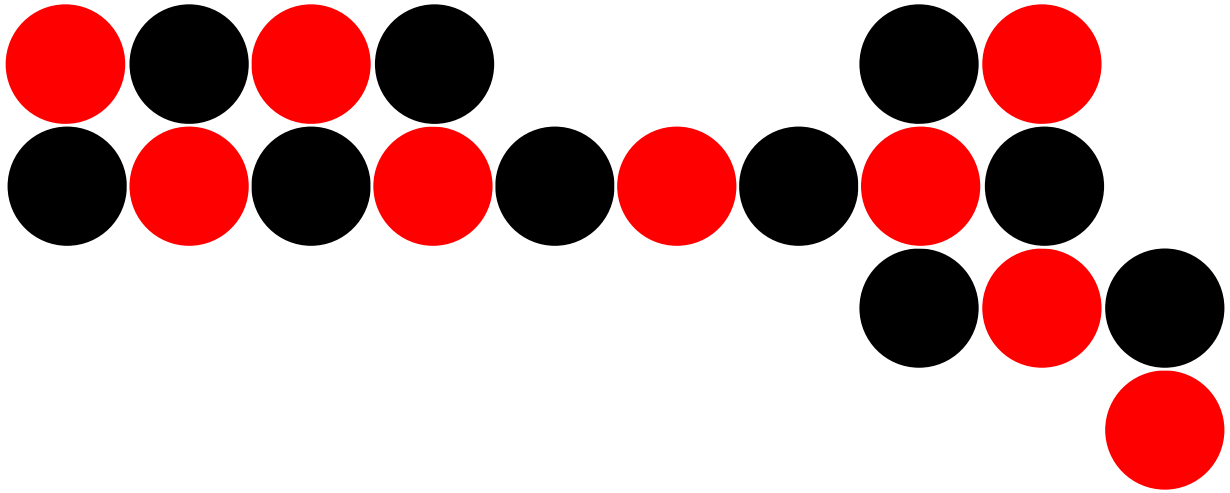
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



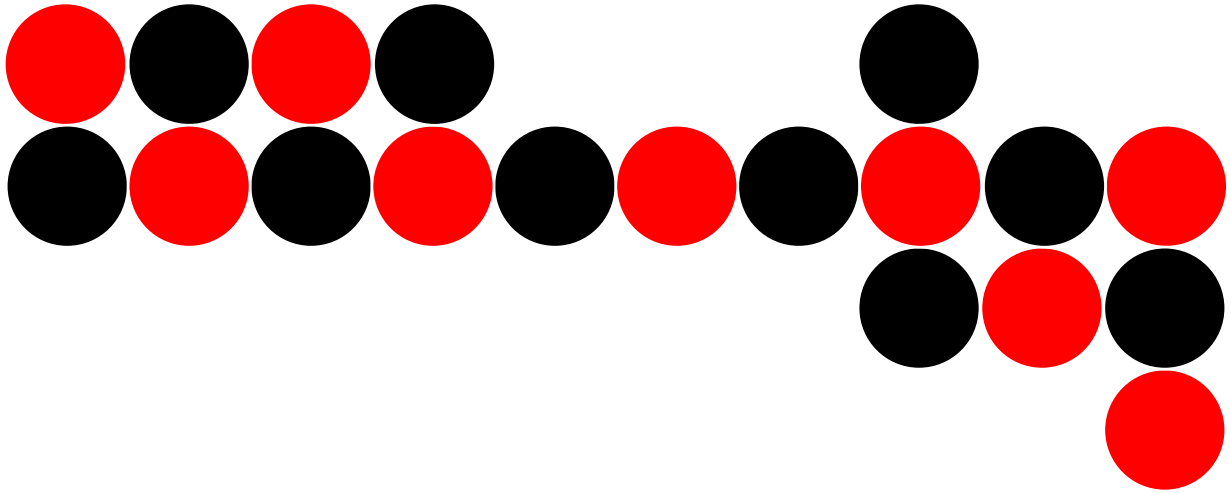
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



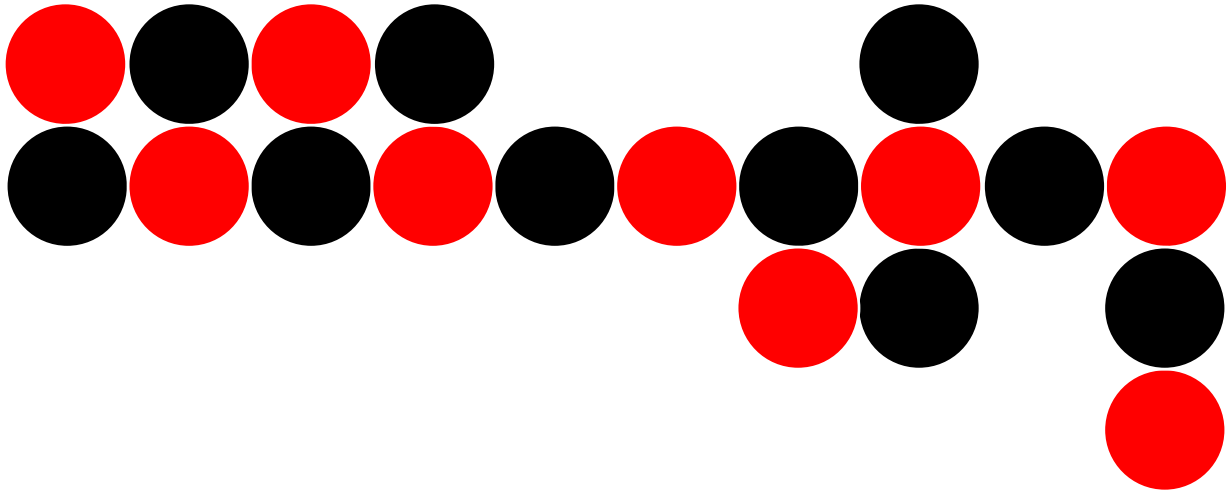
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



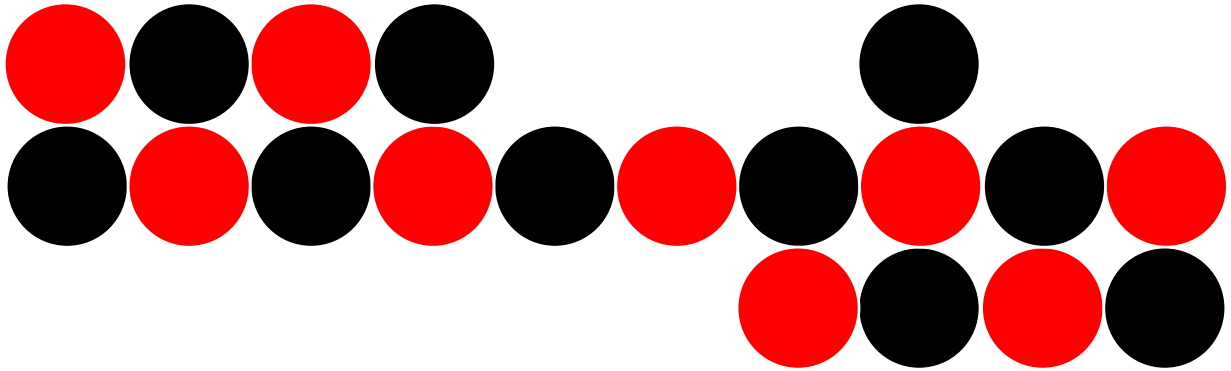
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



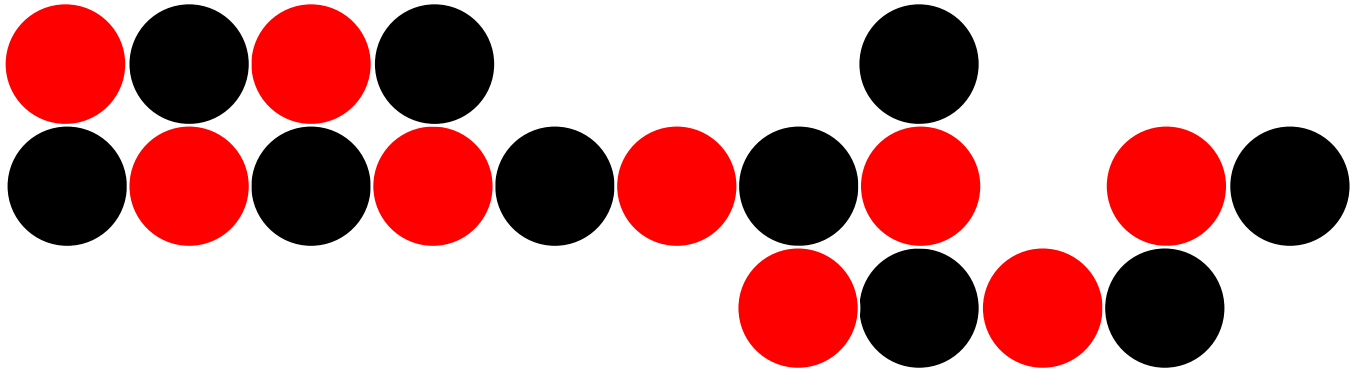
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



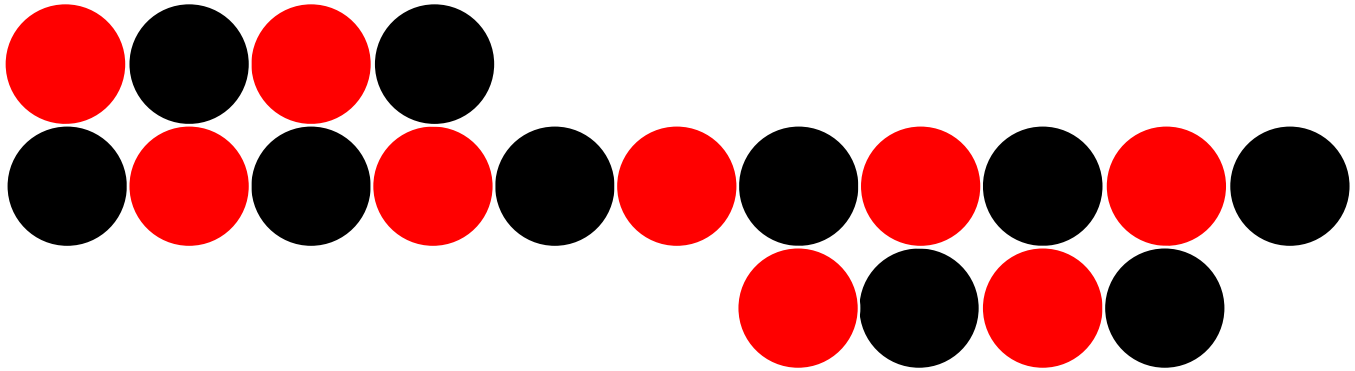
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



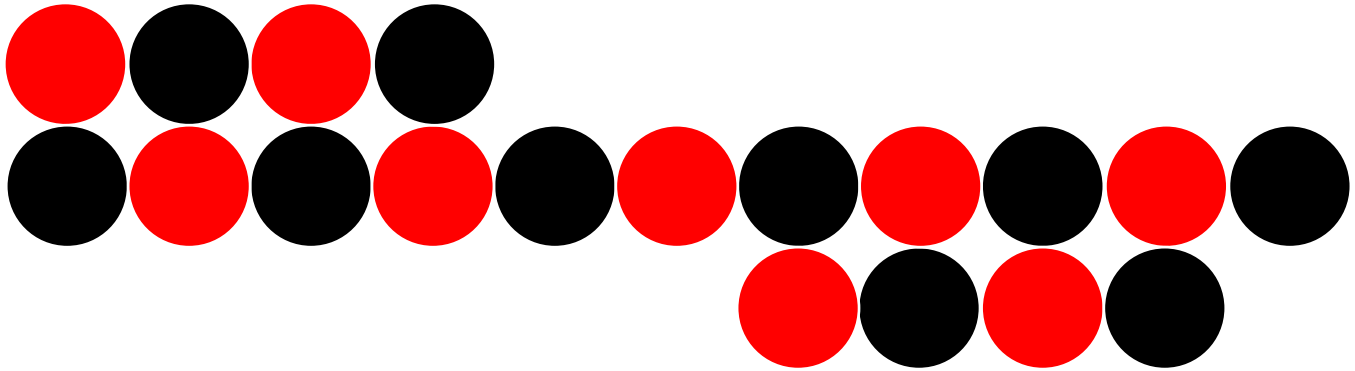
Line to nice shape

Step 2 – Place nodes on other side of the line to act as a mirrored version of the seed



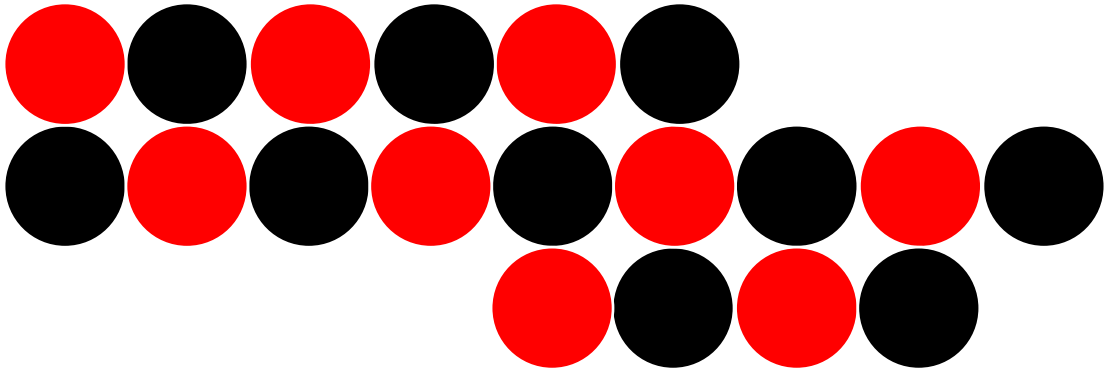
Line to nice shape

Step 3 – Place nodes on the line



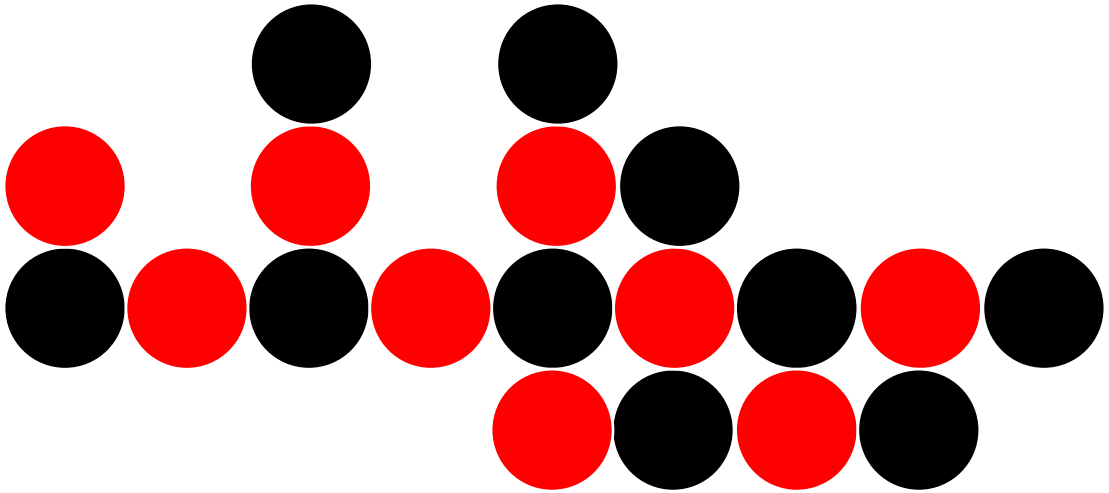
Line to nice shape

Step 3 – Place nodes on the line



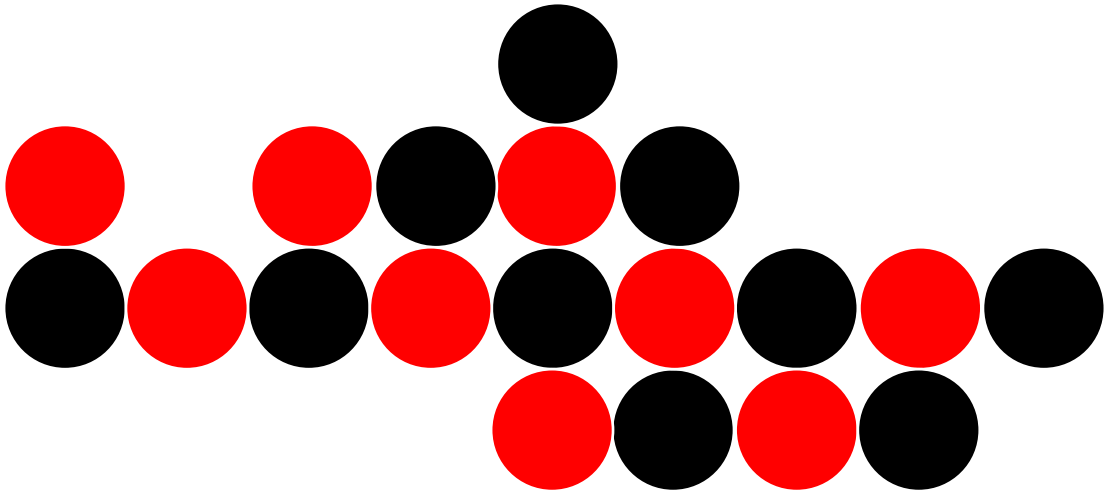
Line to nice shape

Step 3 – Place nodes on the line



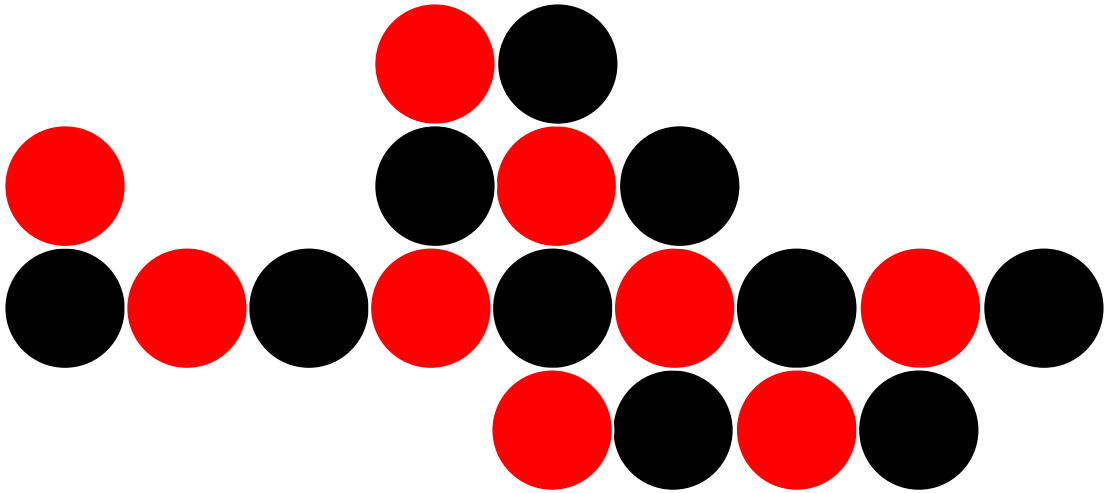
Line to nice shape

Step 3 – Place nodes on the line



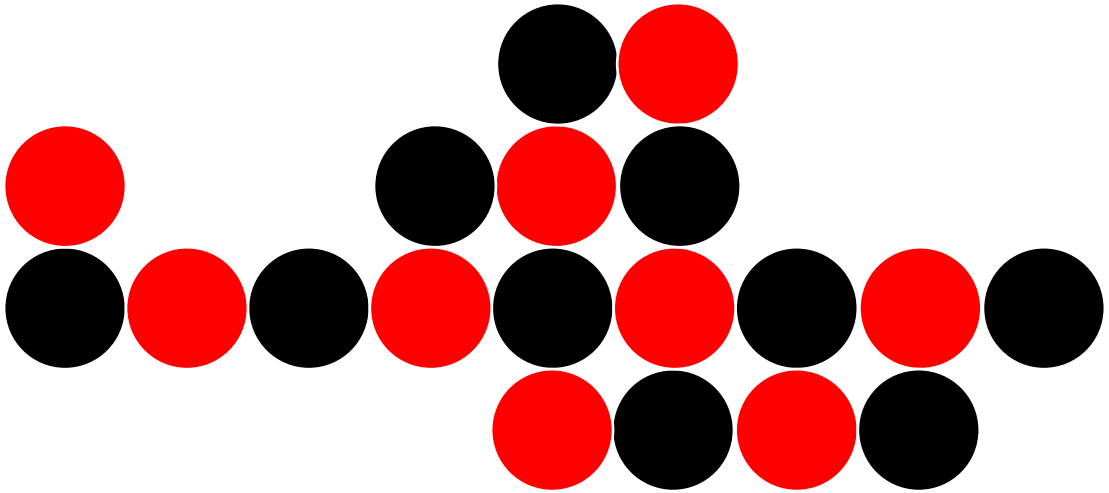
Line to nice shape

Step 3 – Place nodes on the line



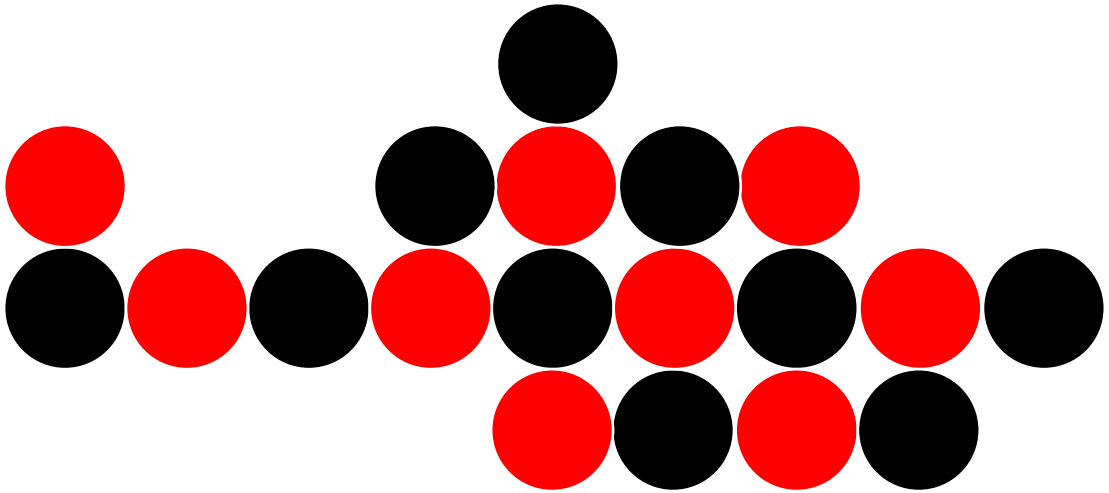
Line to nice shape

Step 3 – Place nodes on the line



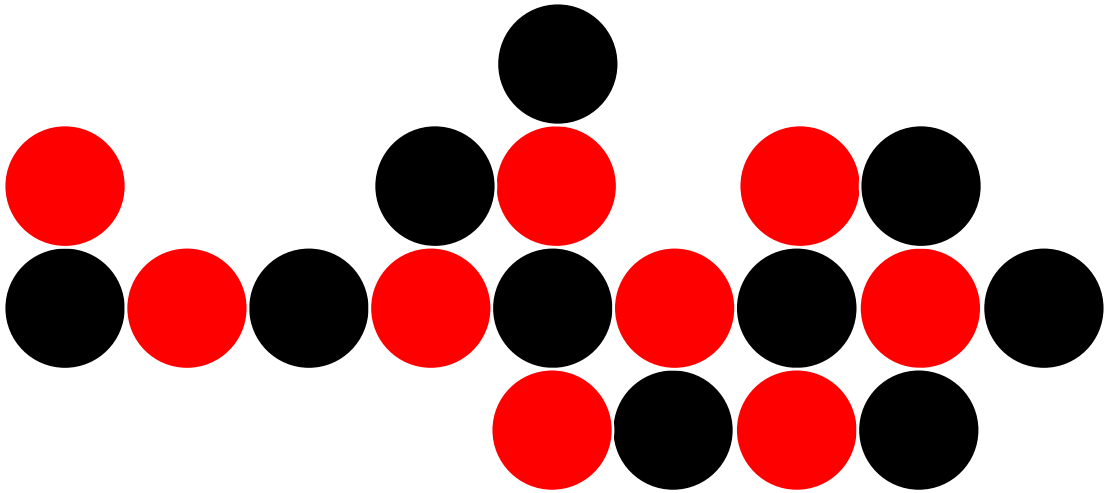
Line to nice shape

Step 3 – Place nodes on the line



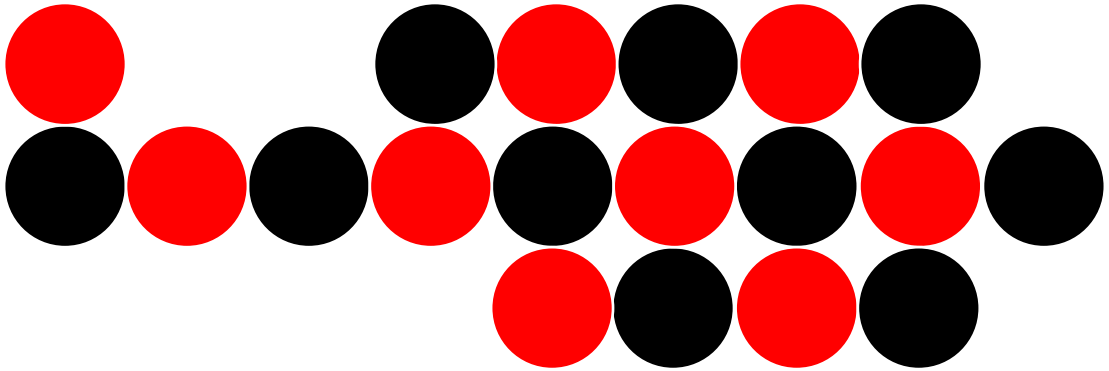
Line to nice shape

Step 3 – Place nodes on the line



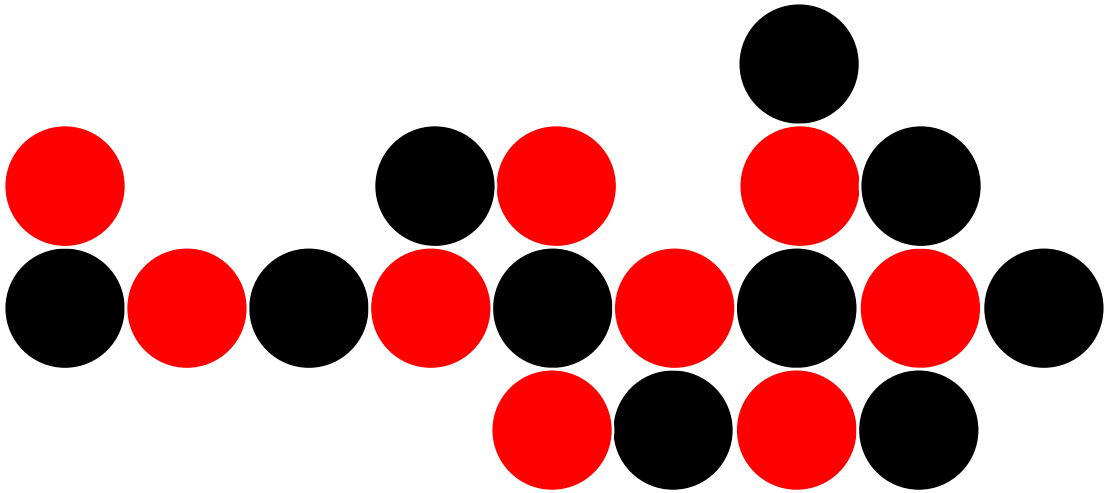
Line to nice shape

Step 3 – Place nodes on the line



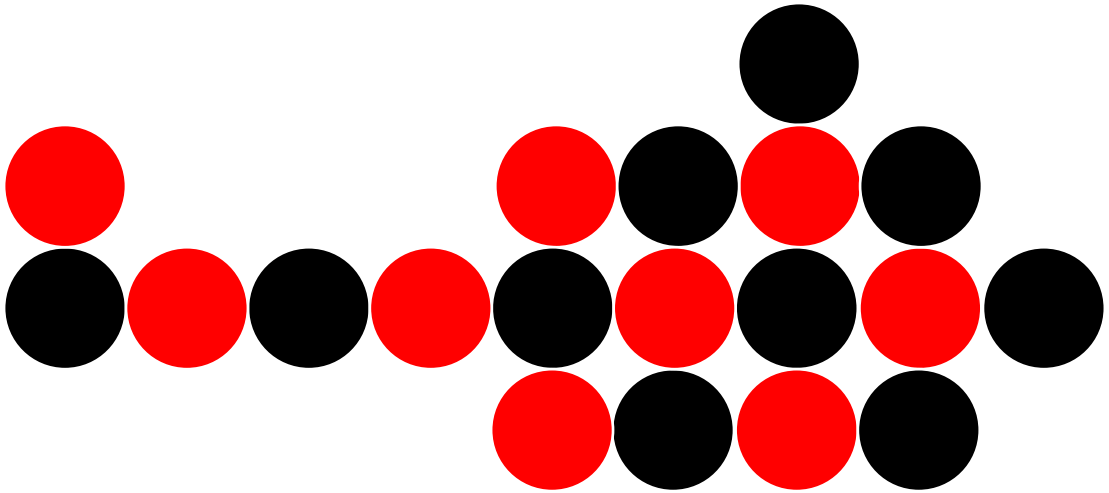
Line to nice shape

Step 3 – Place nodes on the line



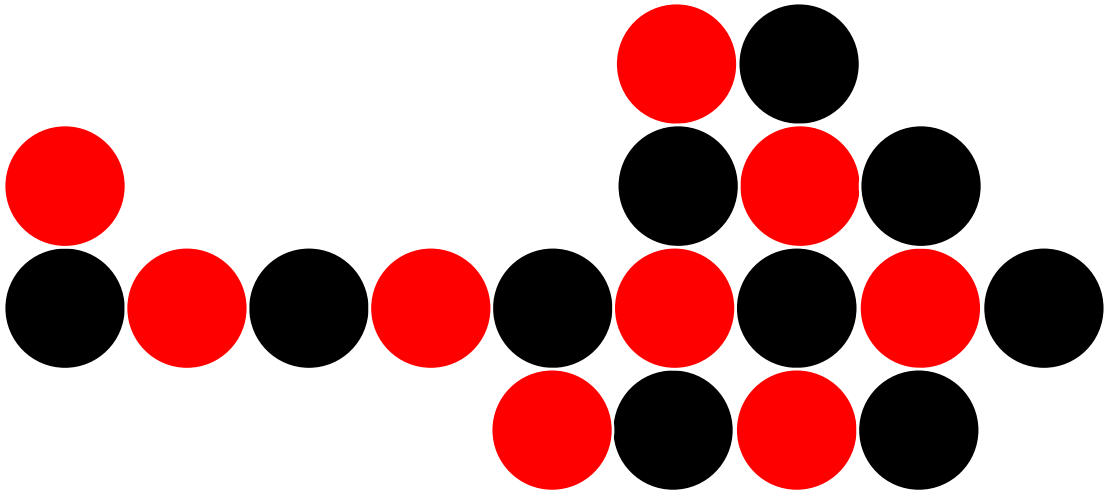
Line to nice shape

Step 3 – Place nodes on the line



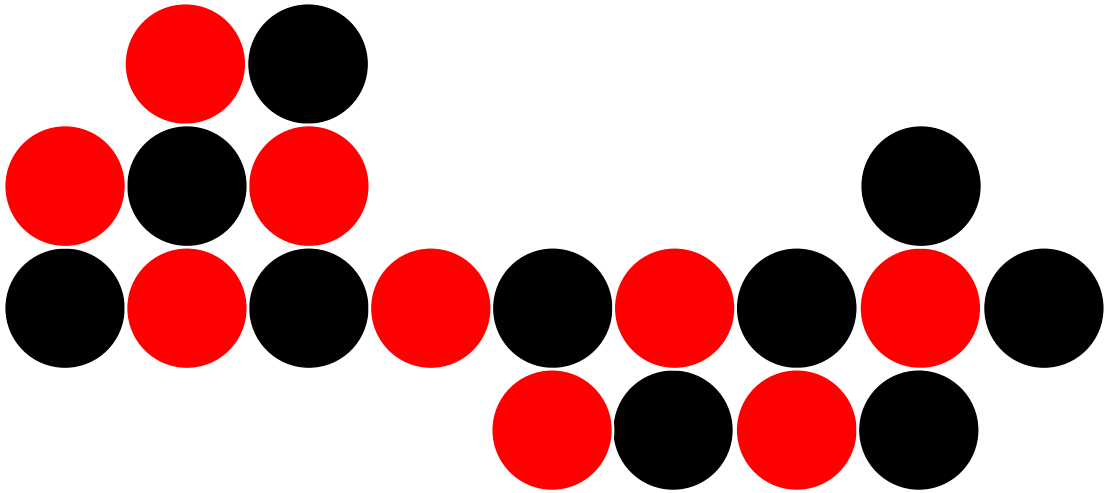
Line to nice shape

Step 3 – Place nodes on the line



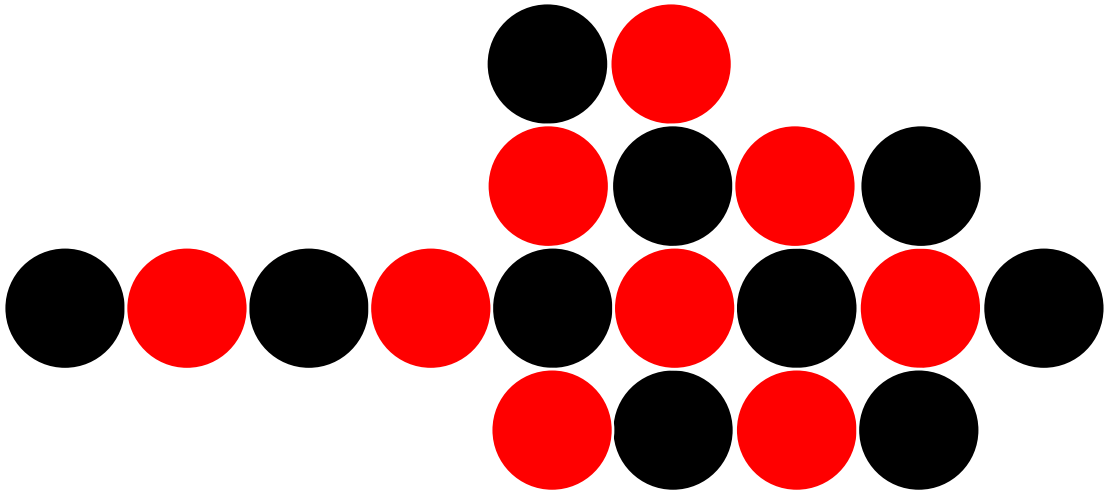
Line to nice shape

Step 3 – Place nodes on the line



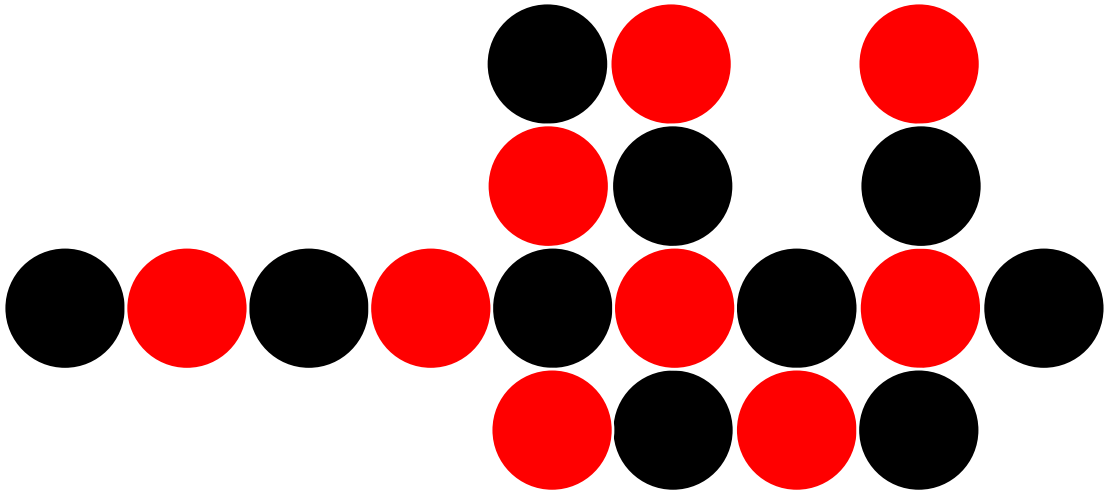
Line to nice shape

Step 3 – Place nodes on the line



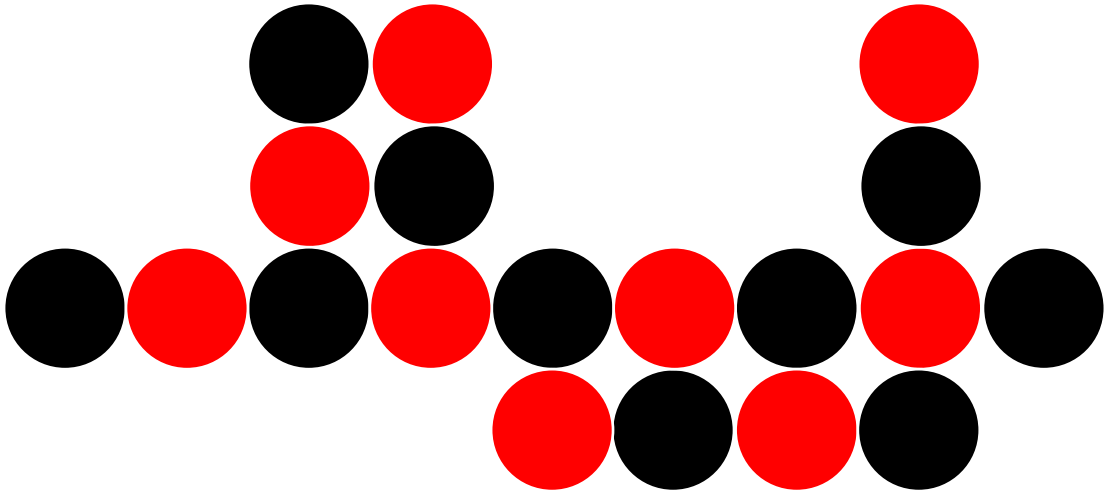
Line to nice shape

Step 3 – Place nodes on the line



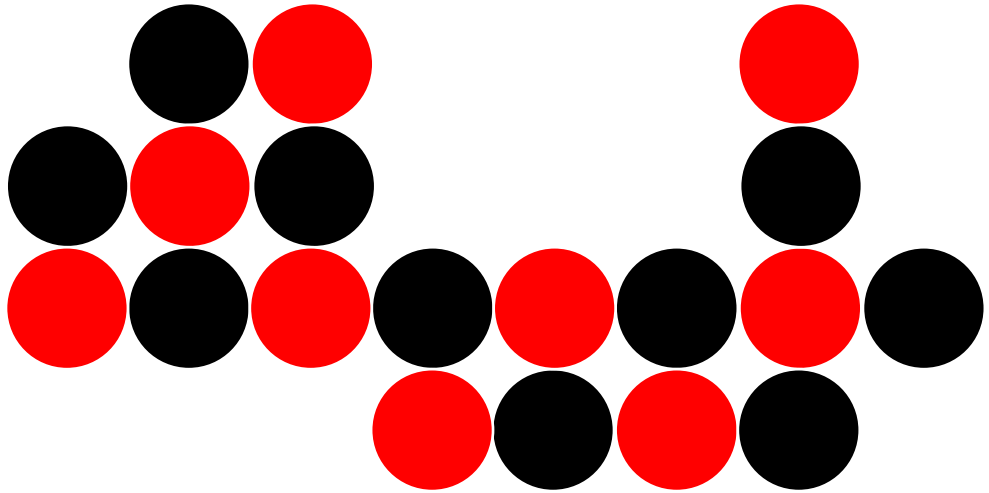
Line to nice shape

Step 3 – Place nodes on the line



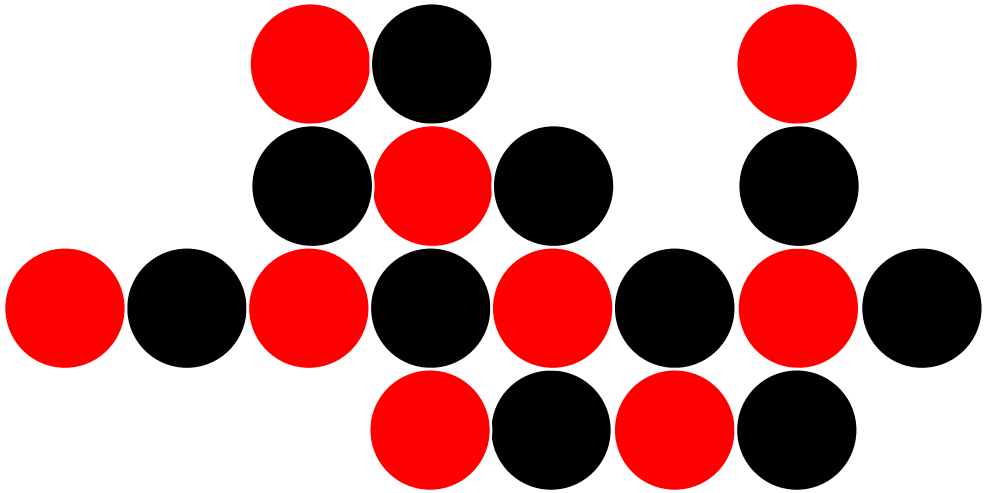
Line to nice shape

Step 3 – Place nodes on the line



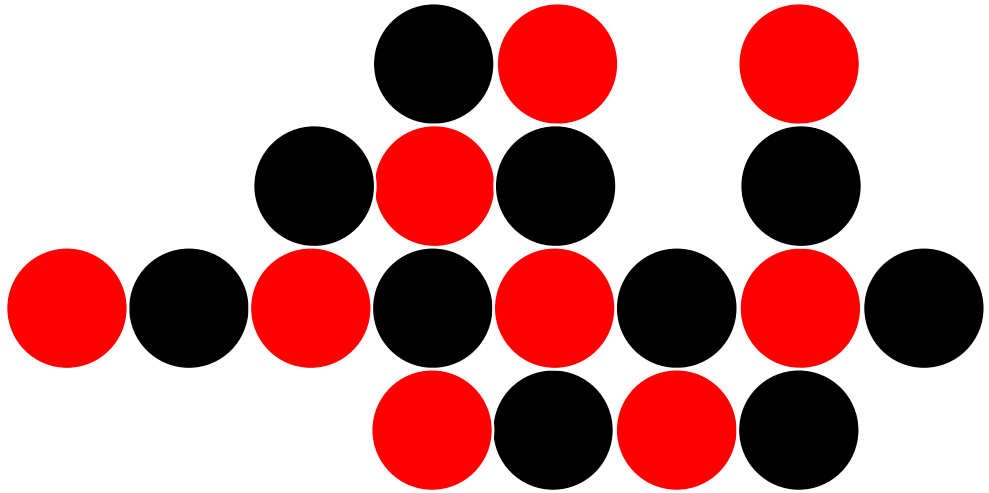
Line to nice shape

Step 3 – Place nodes on the line



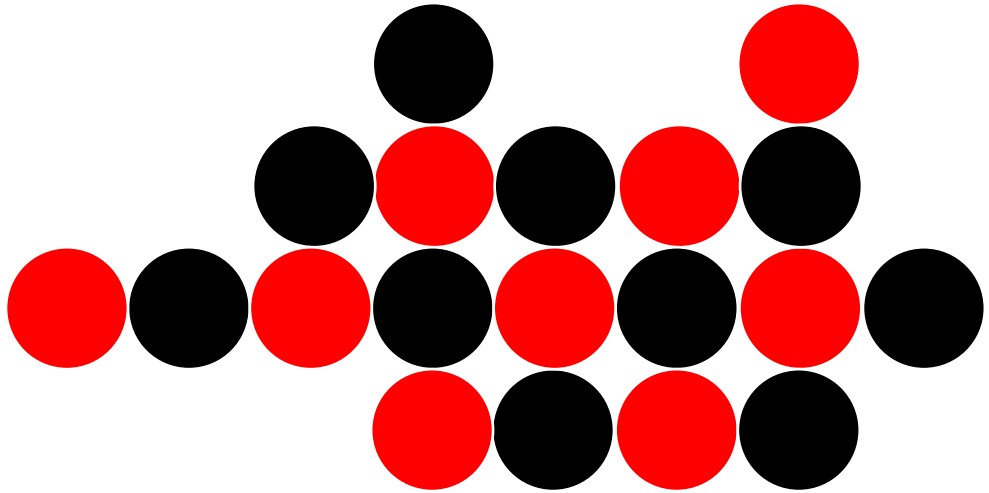
Line to nice shape

Step 3 – Place nodes on the line



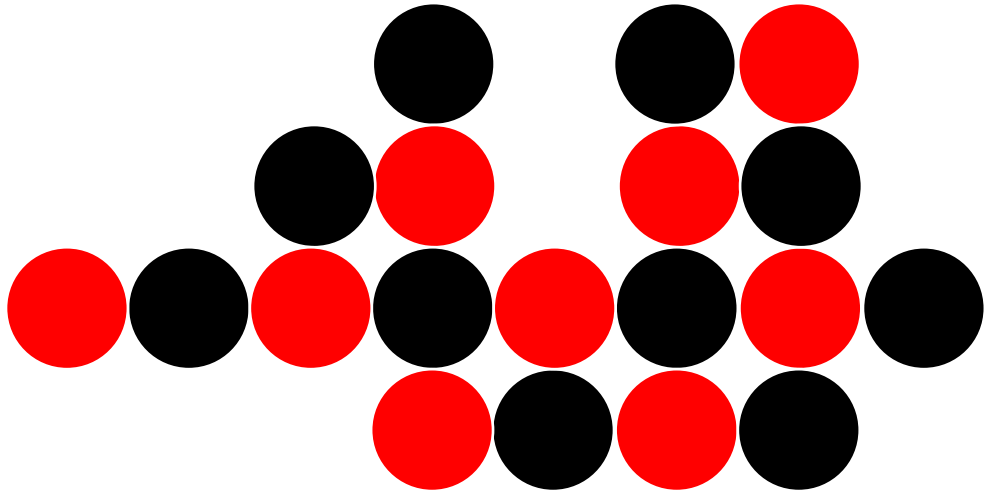
Line to nice shape

Step 3 – Place nodes on the line



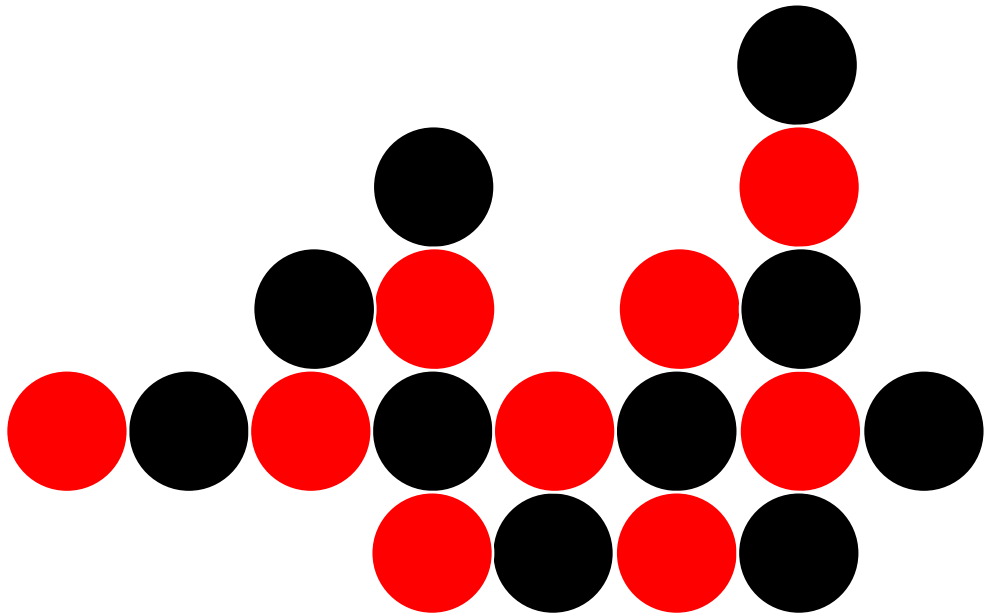
Line to nice shape

Step 3 – Place nodes on the line



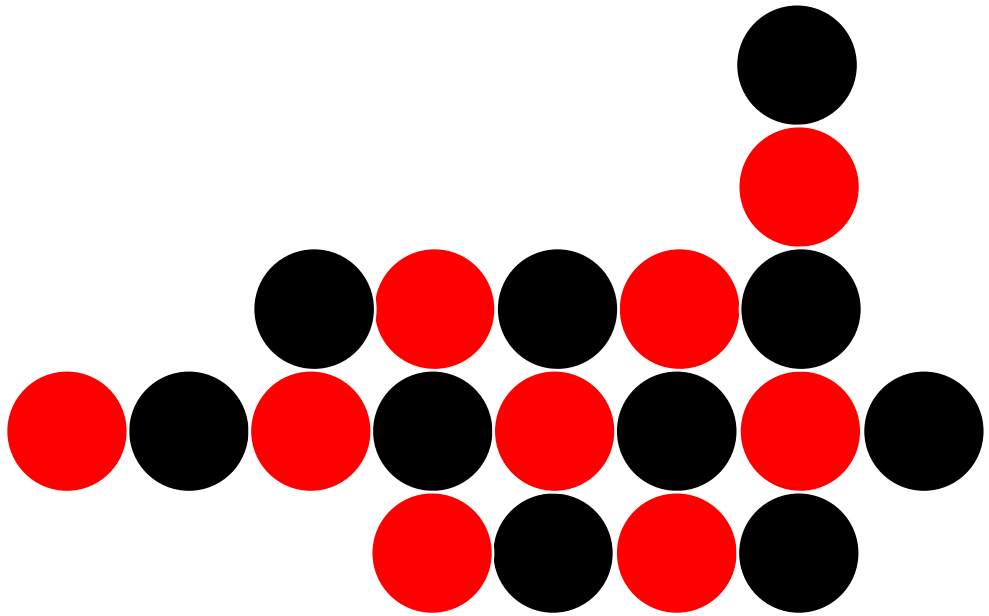
Line to nice shape

Step 3 – Place nodes on the line



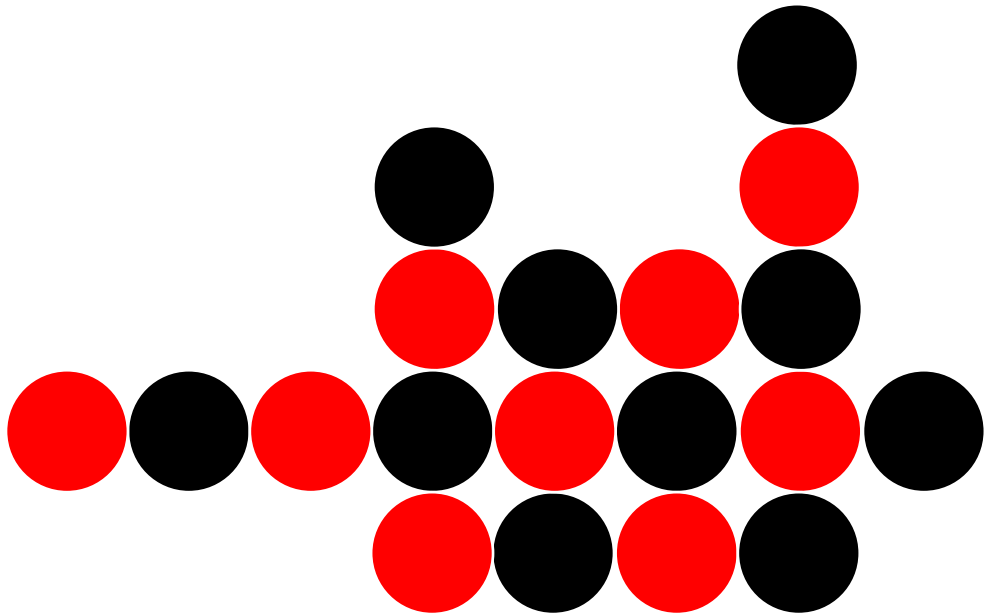
Line to nice shape

Step 3 – Place nodes on the line



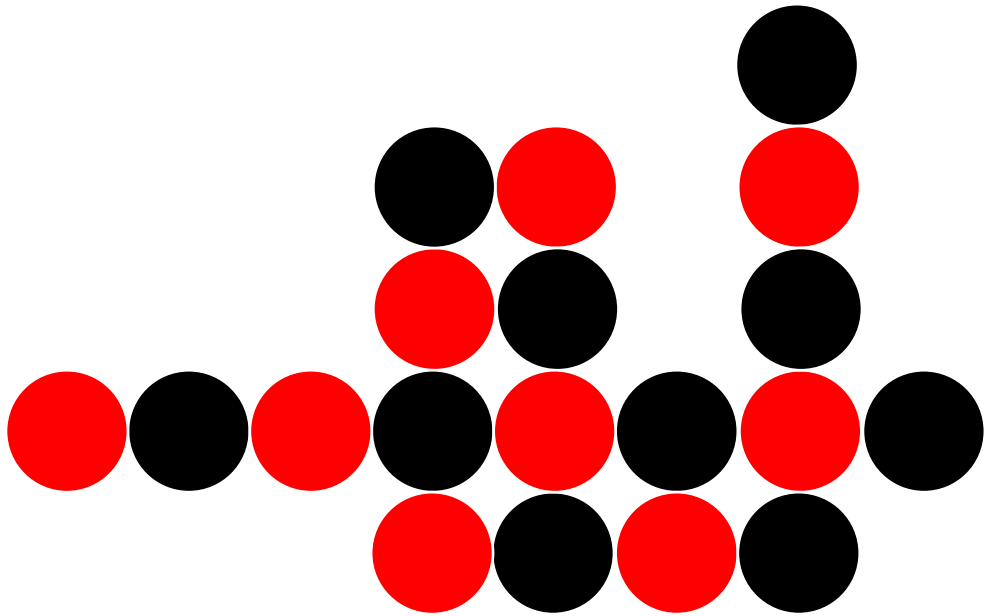
Line to nice shape

Step 3 – Place nodes on the line



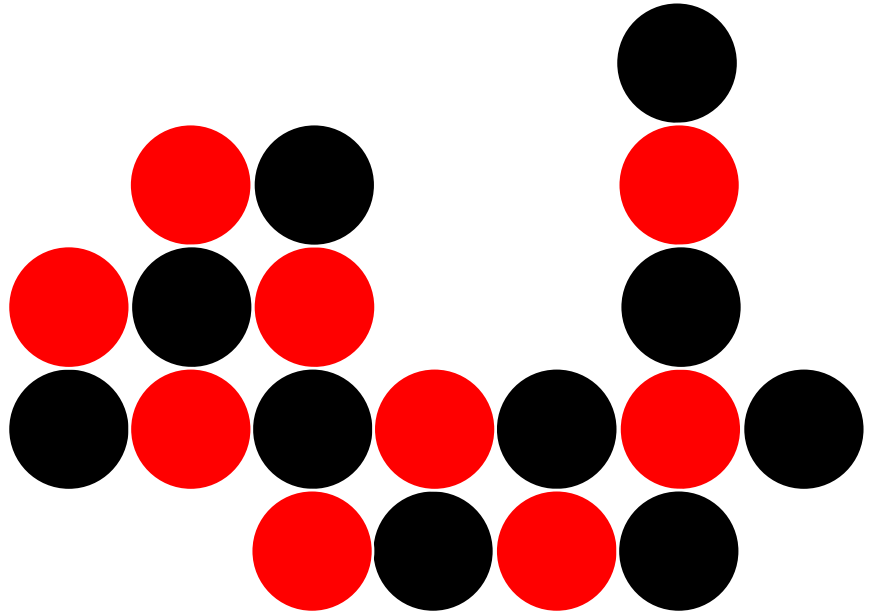
Line to nice shape

Step 3 – Place nodes on the line



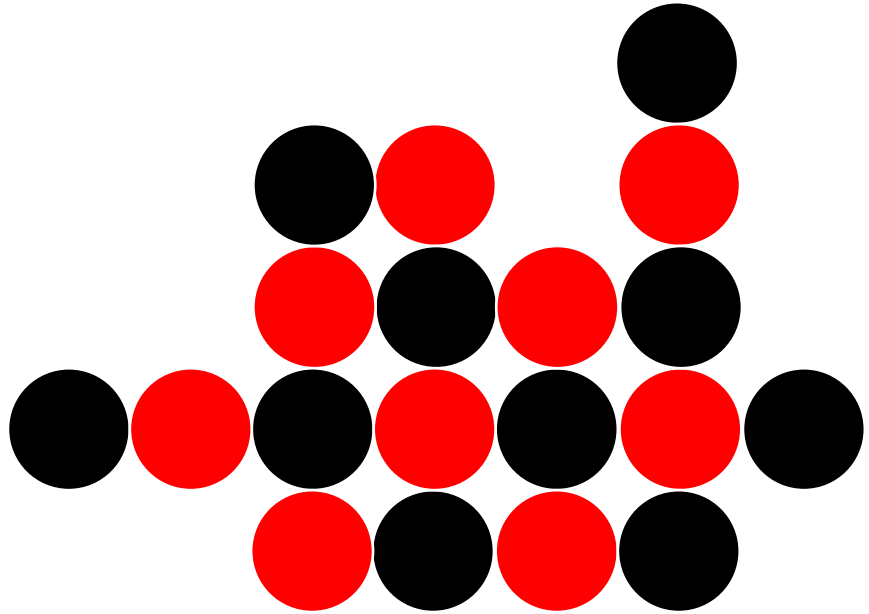
Line to nice shape

Step 3 – Place nodes on the line



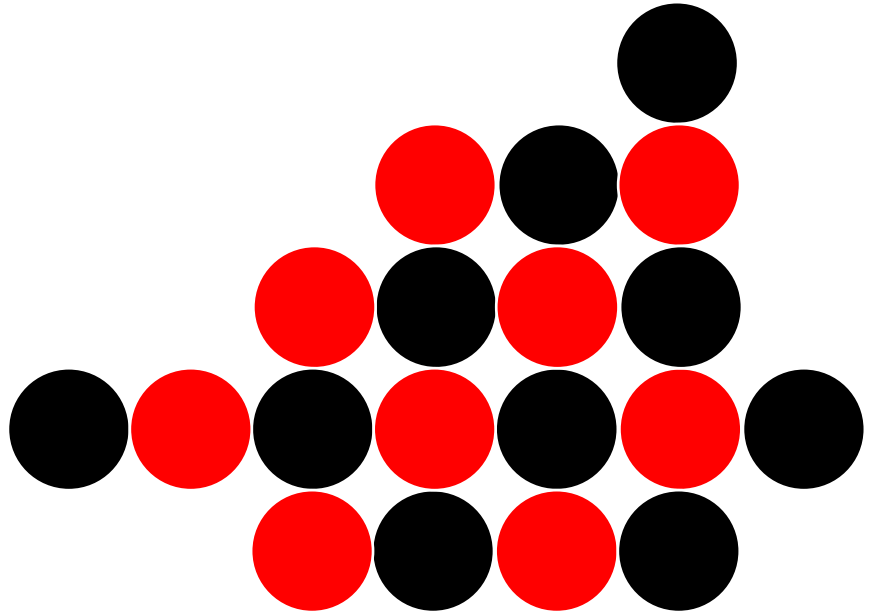
Line to nice shape

Step 3 – Place nodes on the line



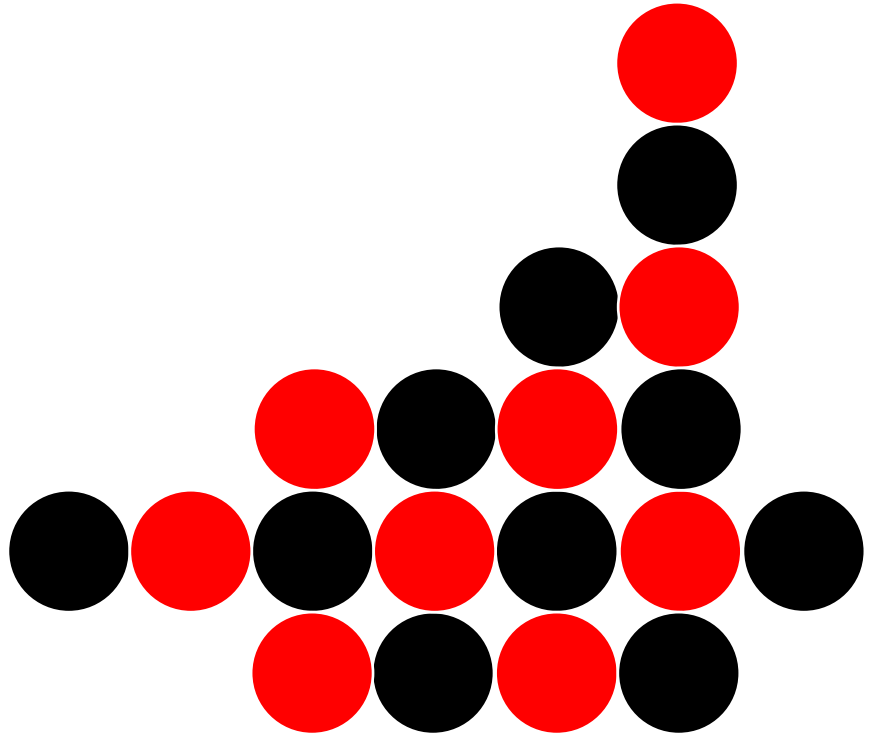
Line to nice shape

Step 3 – Place nodes on the line



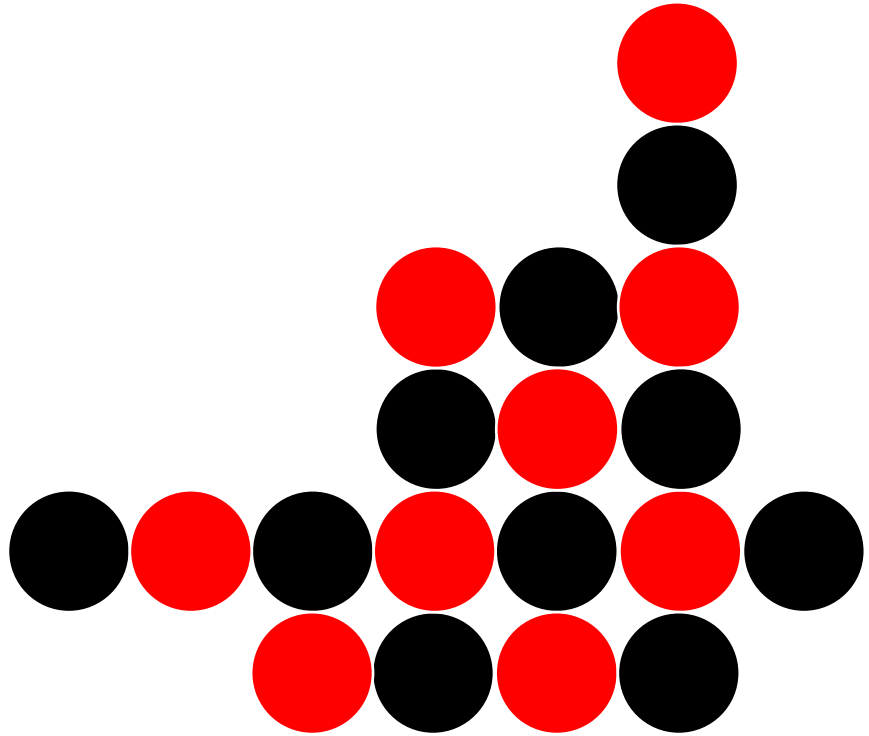
Line to nice shape

Step 3 – Place nodes on the line



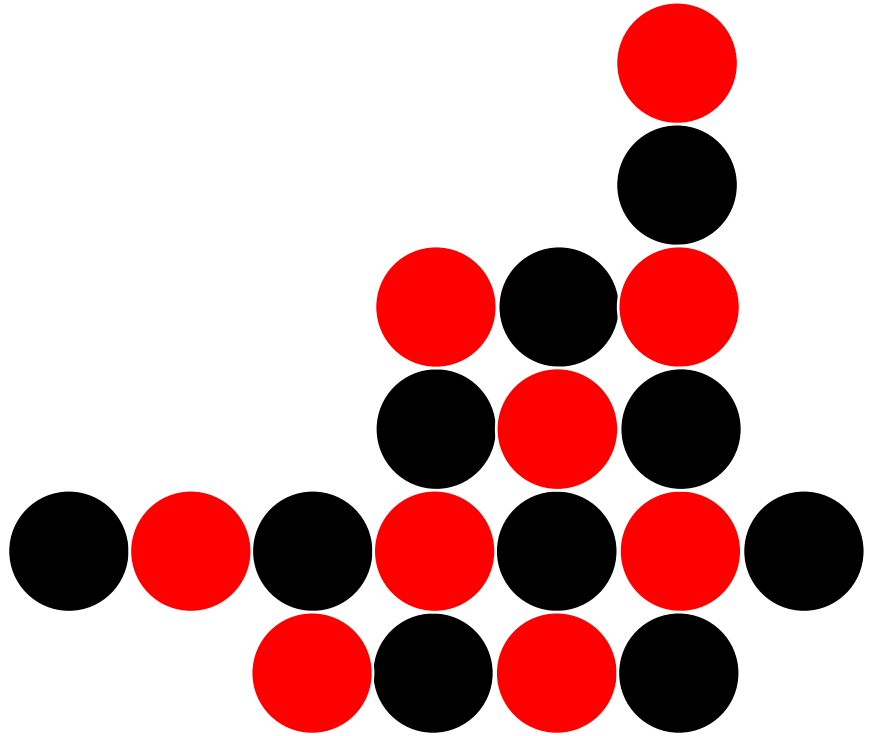
Line to nice shape

Step 3 – Place nodes on the line



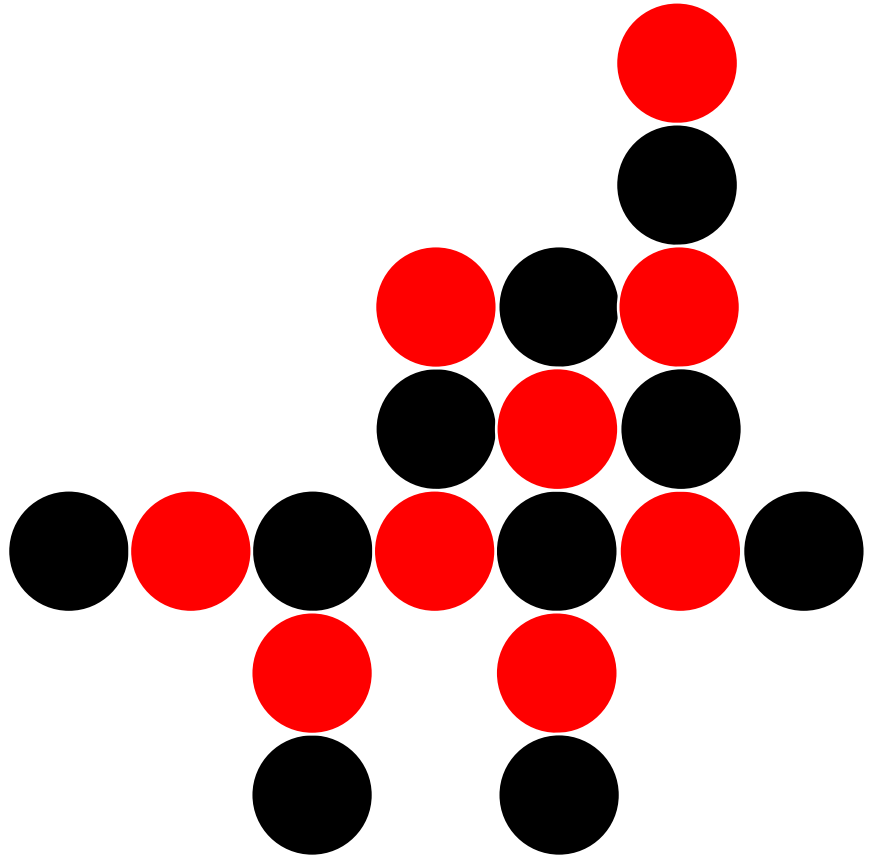
Line to nice shape

Step 4 – Place nodes of the mirrored seed



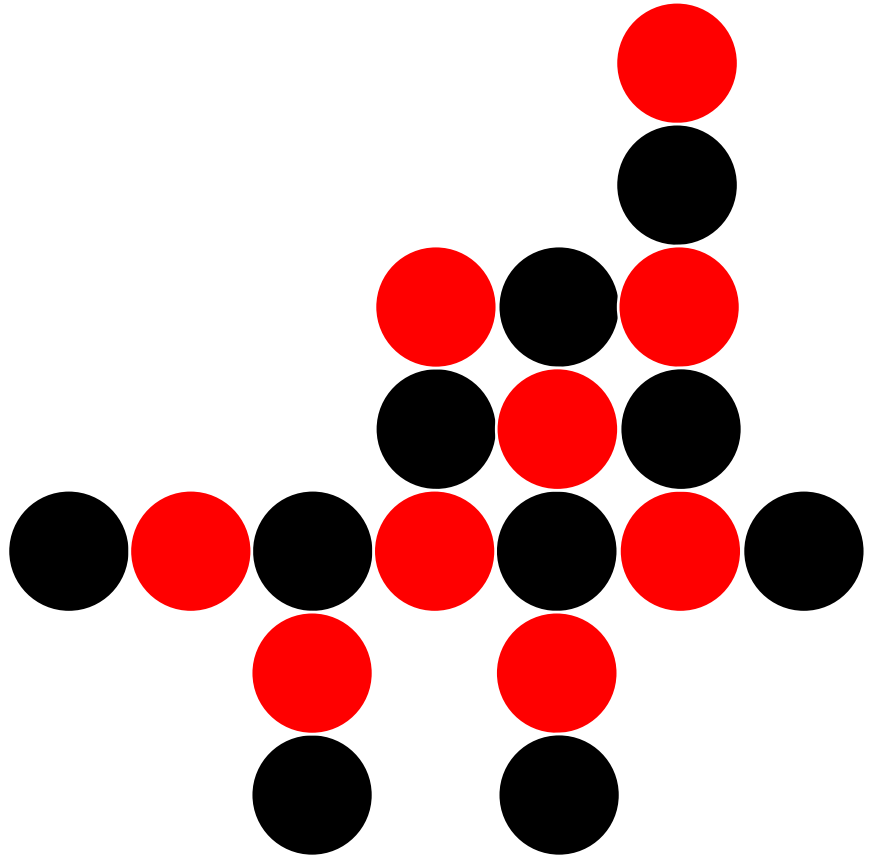
Line to nice shape

Step 4 – Place nodes of the mirrored seed



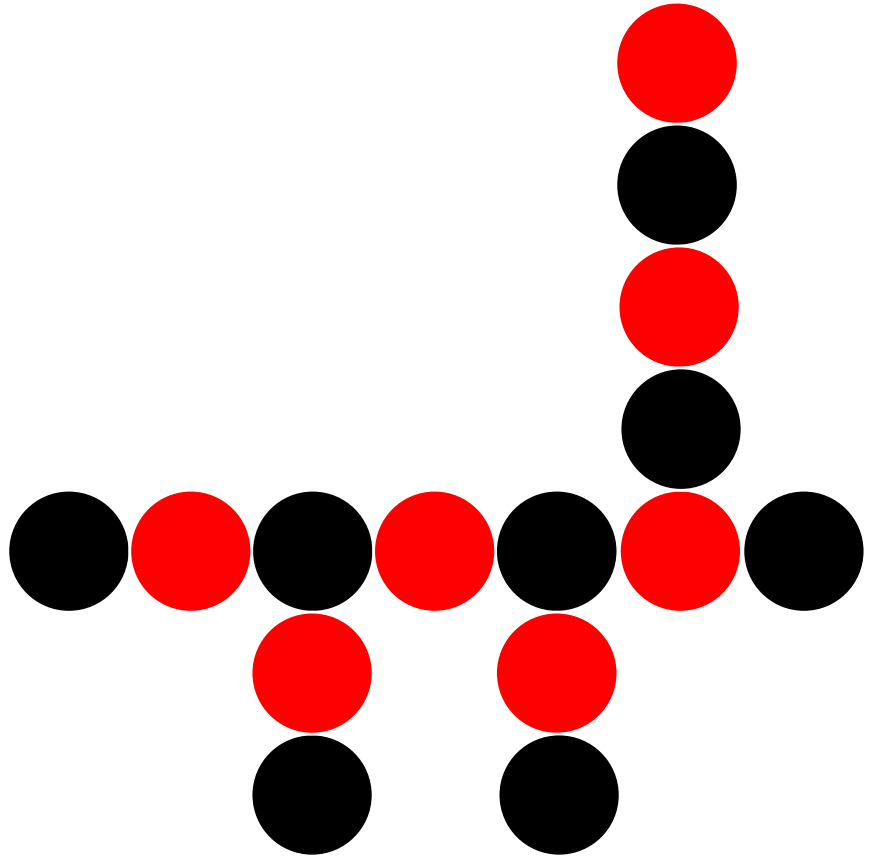
Line to nice shape

Step 5 – discard nodes of the seed + 1



Line to nice shape

Step 5 – discard nodes of the seed + 1



Nice shape to nice shape

Line to nice shape – 3 node seed, 1 node waste

By reversibility nice shape to line – 4 node seed

The line becomes a **canonical shape** between nice shape-nice shape transformations

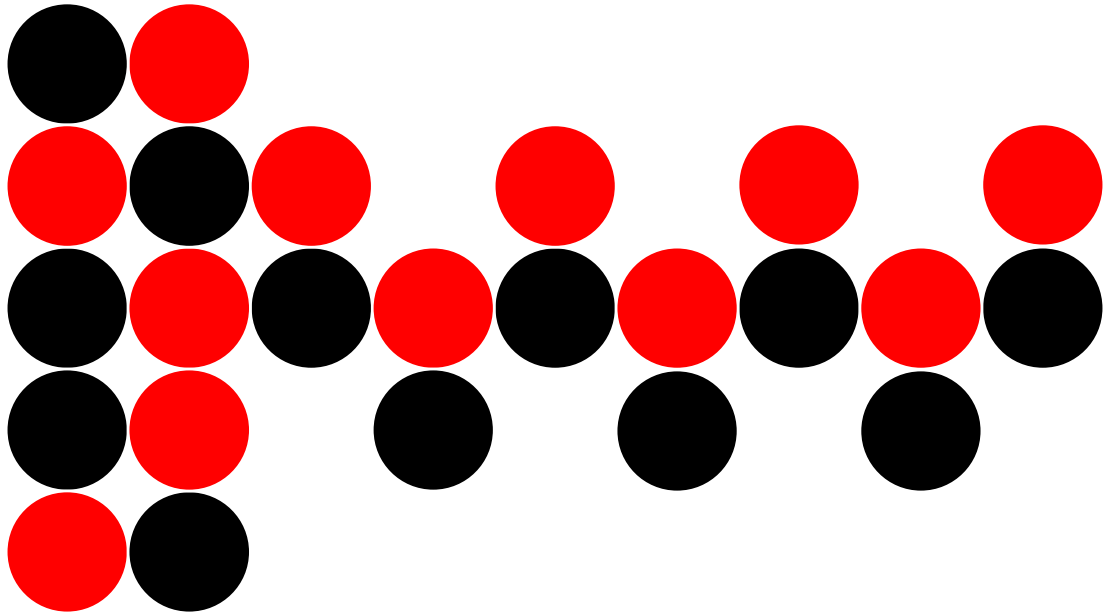
The foundation

Construction relies on switching between colours

If the perpendicular lines are of an **odd** length naïve construction can break this condition

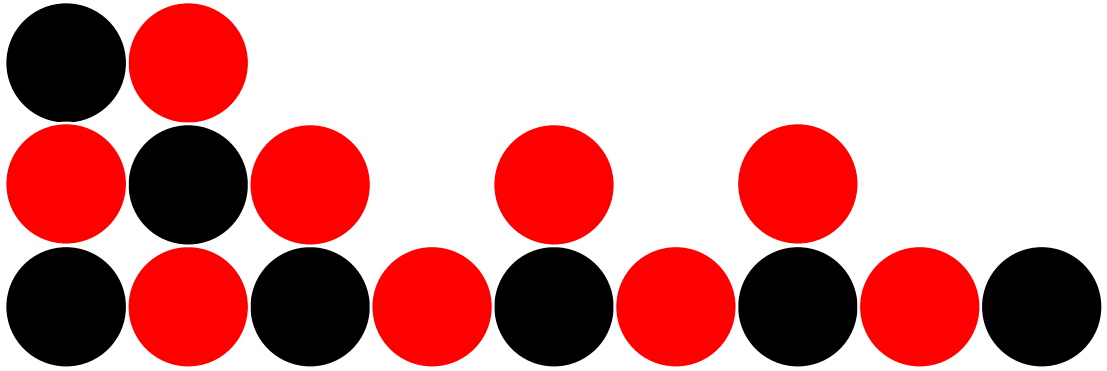
Solution – lay some nodes first as a **foundation**

This foundation is guaranteed to alternate – colour consistency



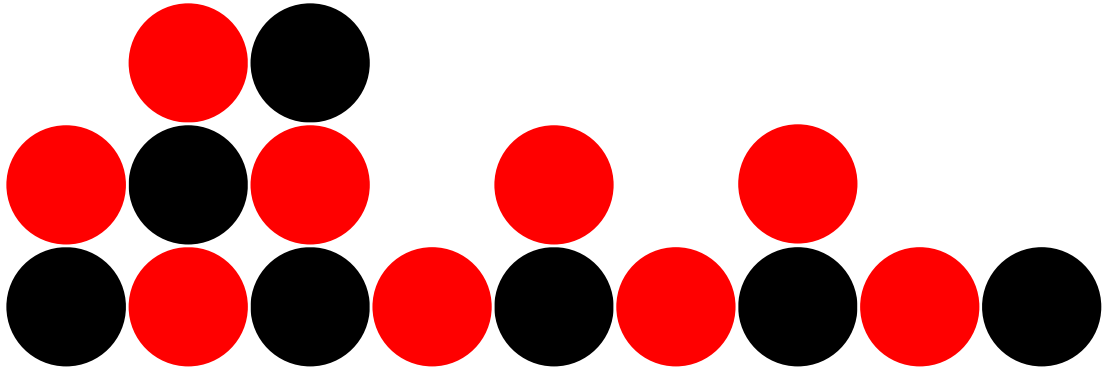
The foundation

Case 1 – same colour



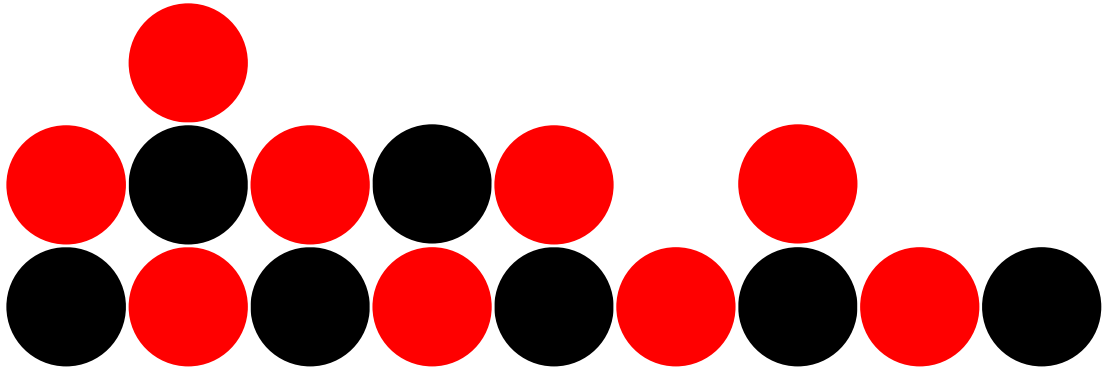
The foundation

Case 1 – same colour



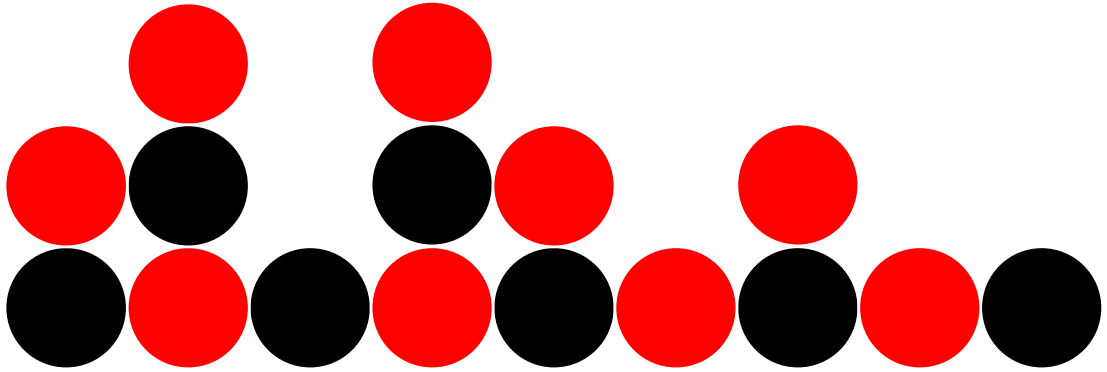
The foundation

Case 1 – same colour



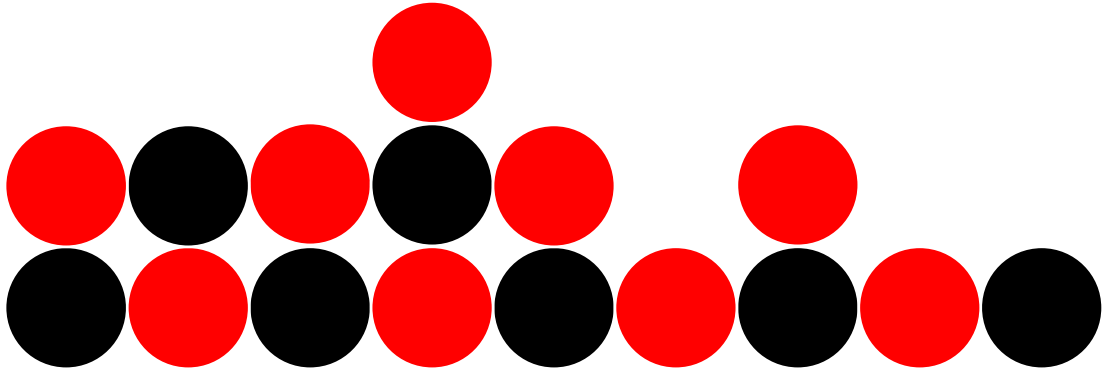
The foundation

Case 1 – same colour



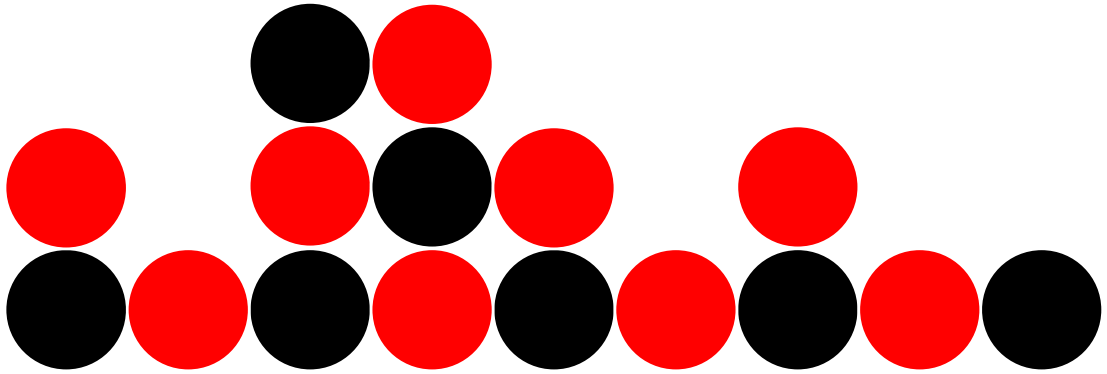
The foundation

Case 1 – same colour



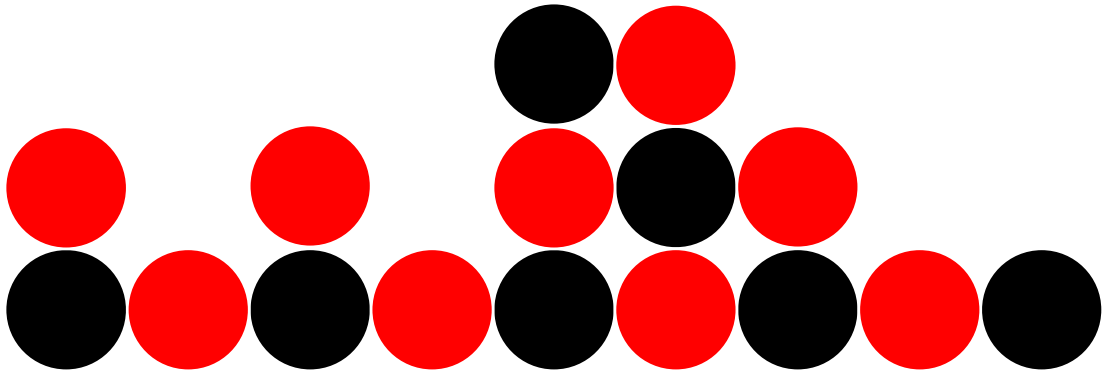
The foundation

Case 1 – same colour



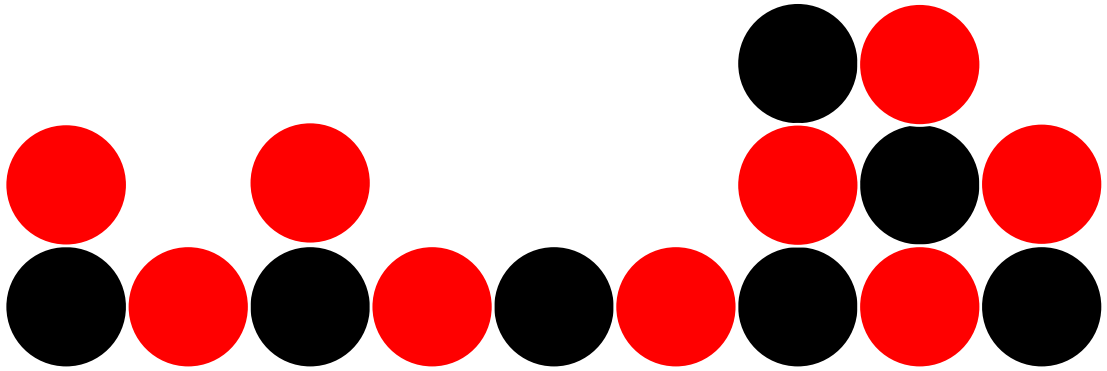
The foundation

Case 1 – same colour



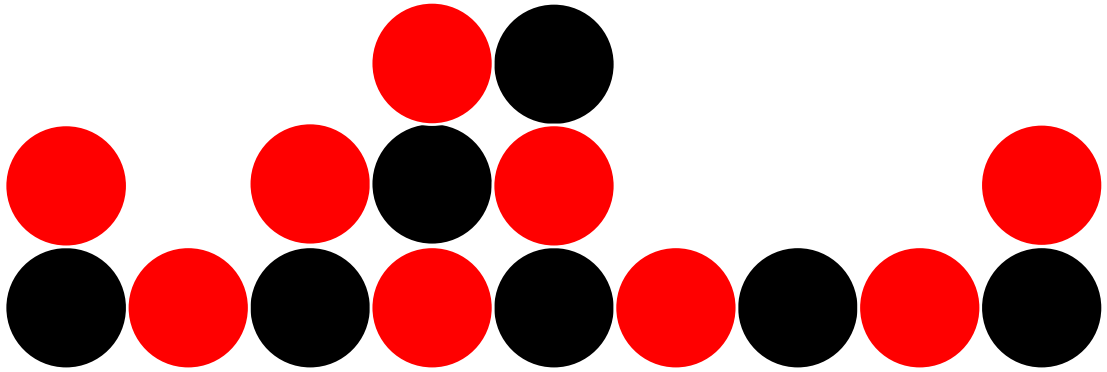
The foundation

Case 1 – same colour



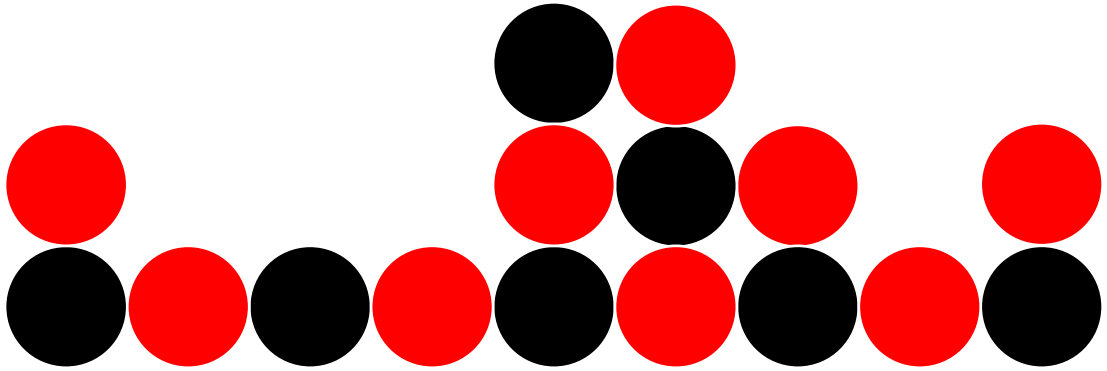
The foundation

Case 1 – same colour



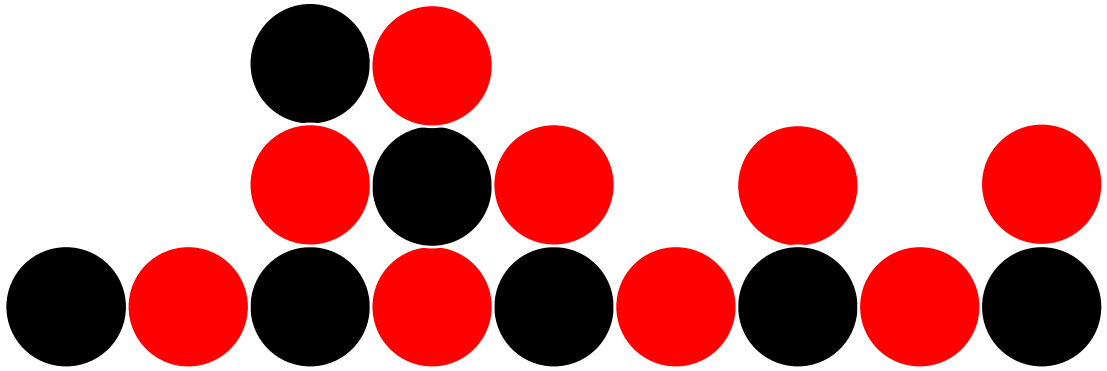
The foundation

Case 1 – same colour



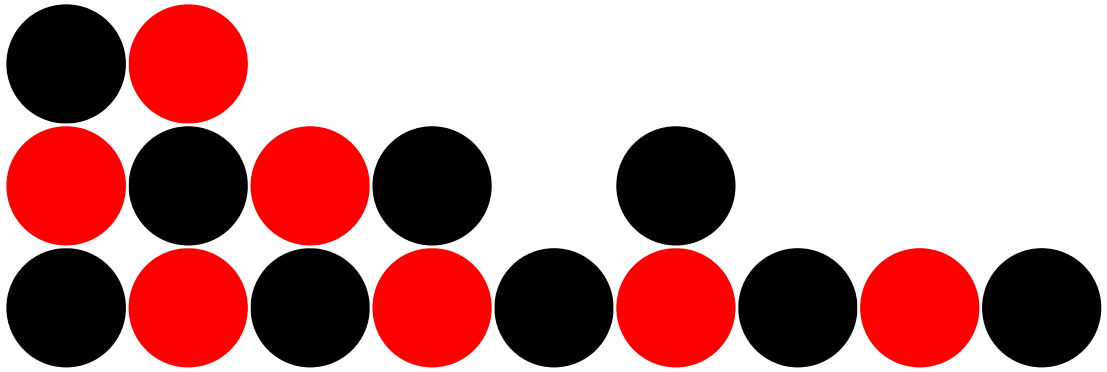
The foundation

Case 1 – same colour



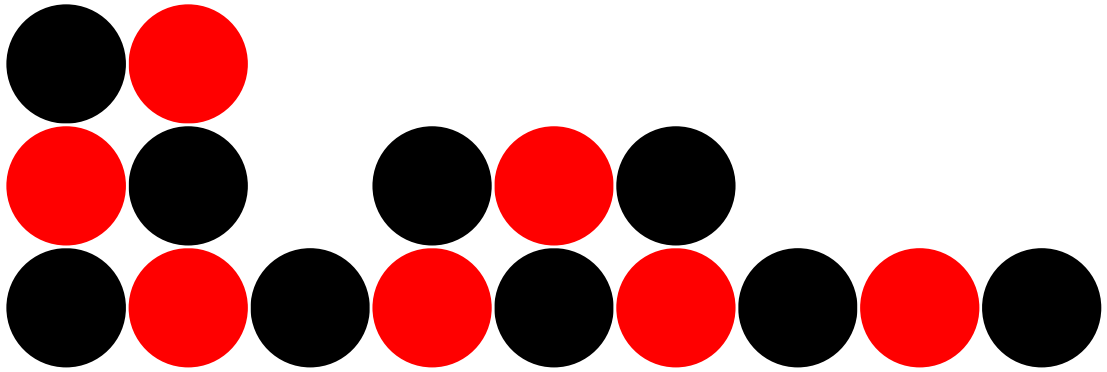
The foundation

Case 2 – different colour



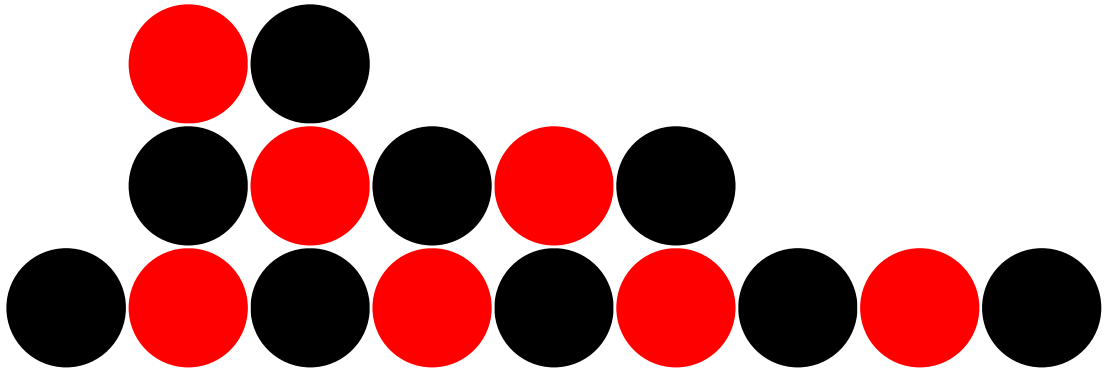
The foundation

Case 2 – different colour



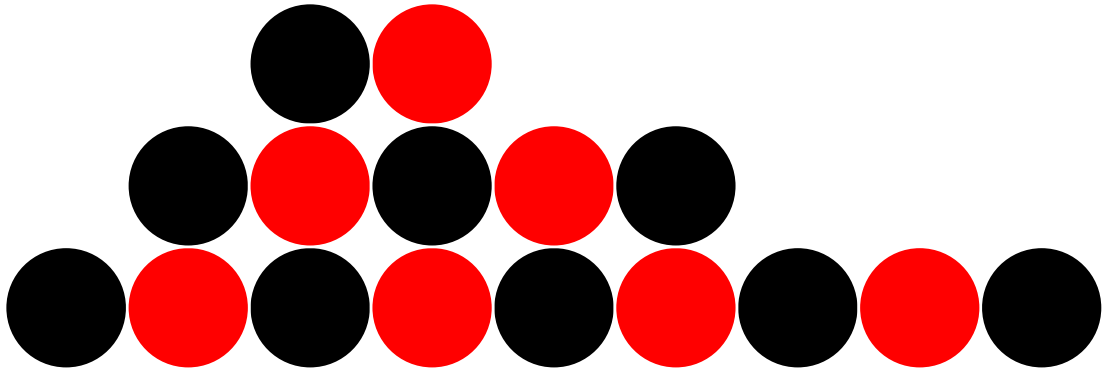
The foundation

Case 2 – different colour



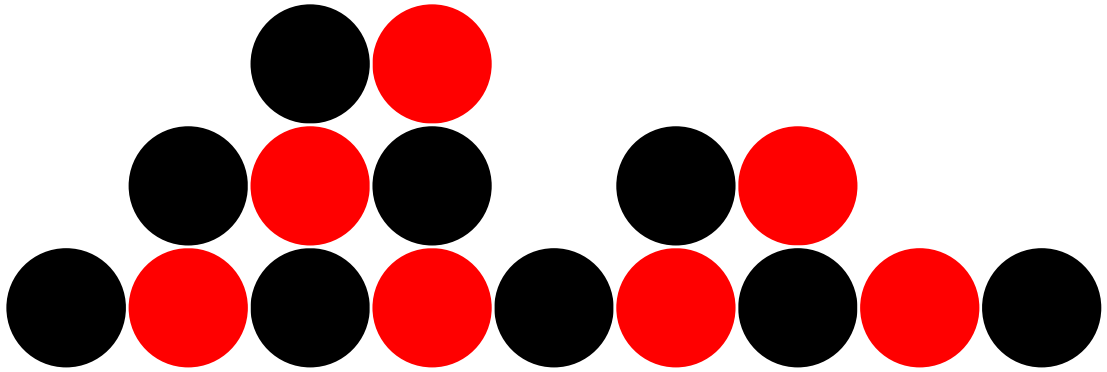
The foundation

Case 2 – different colour



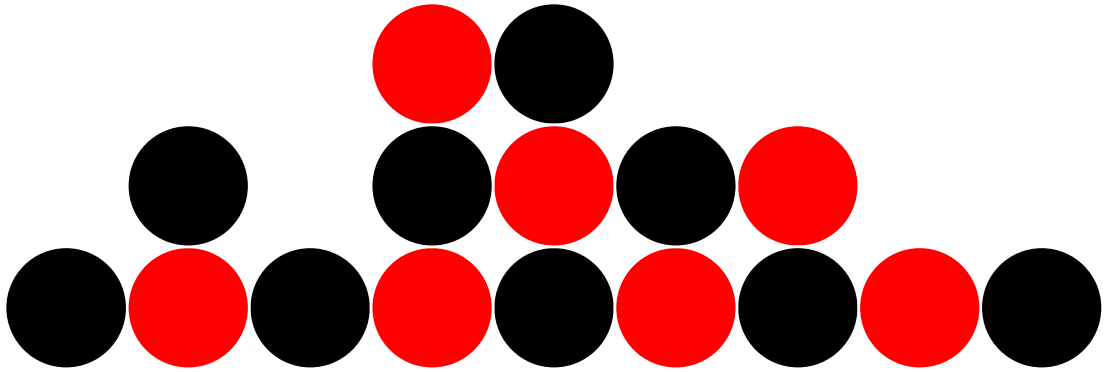
The foundation

Case 2 – different colour



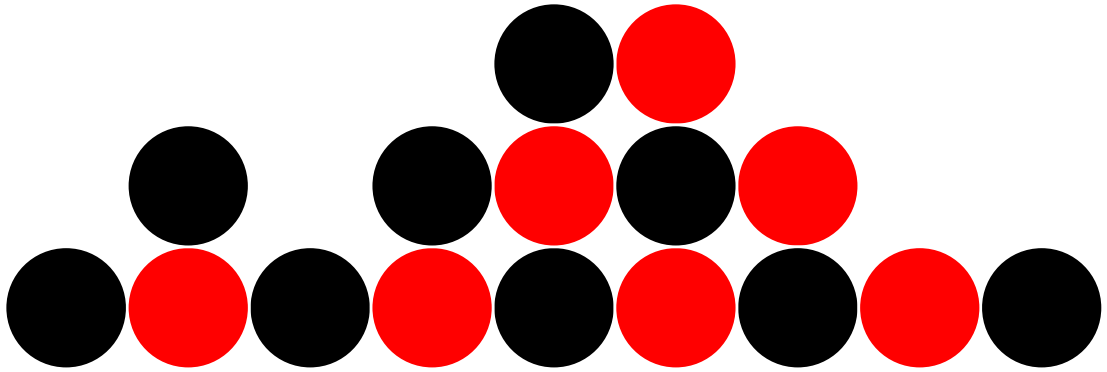
The foundation

Case 2 – different colour



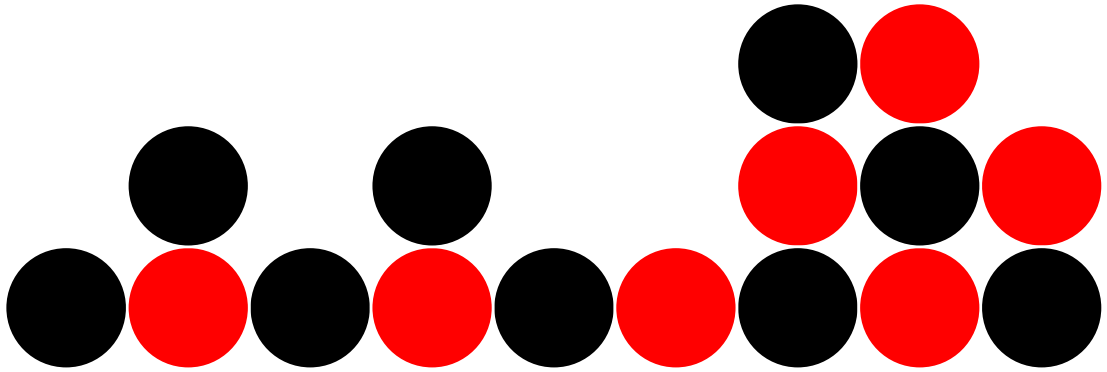
The foundation

Case 2 – different colour



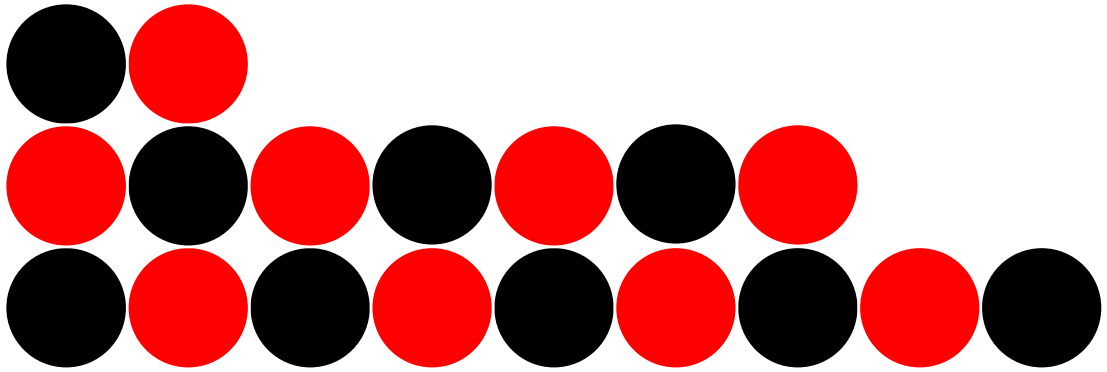
The foundation

Case 2 – different colour



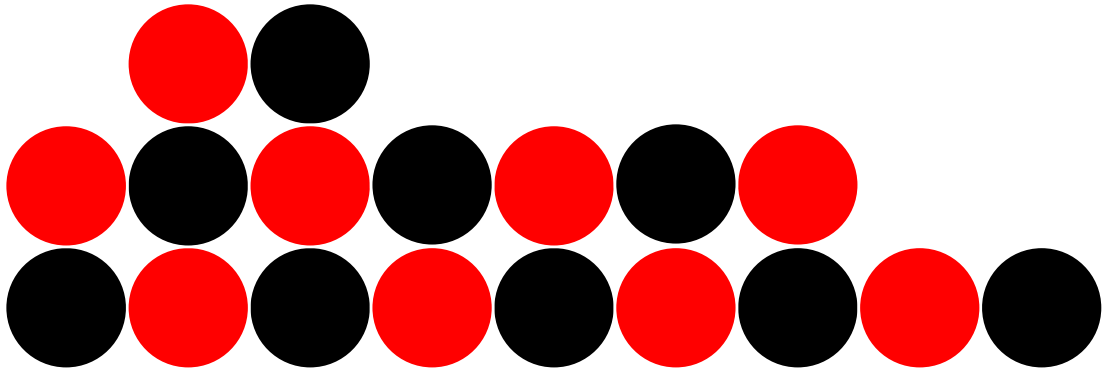
The foundation

Case 3 – chain of nodes



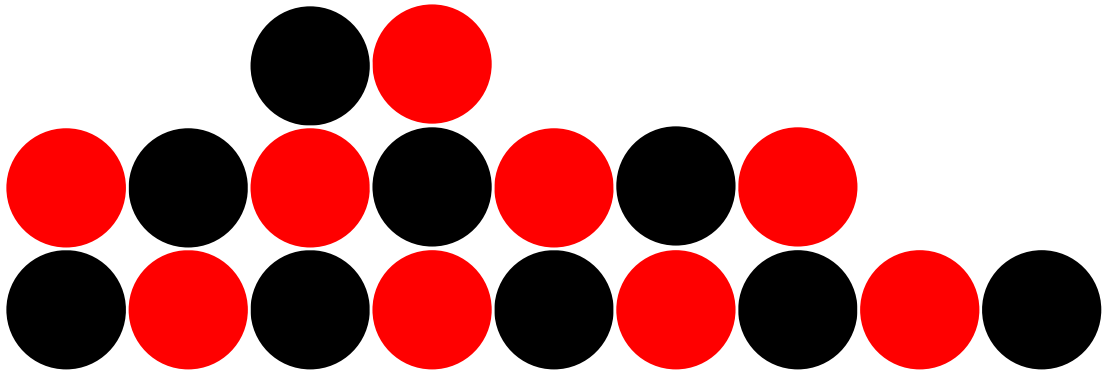
The foundation

Case 3 – chain of nodes



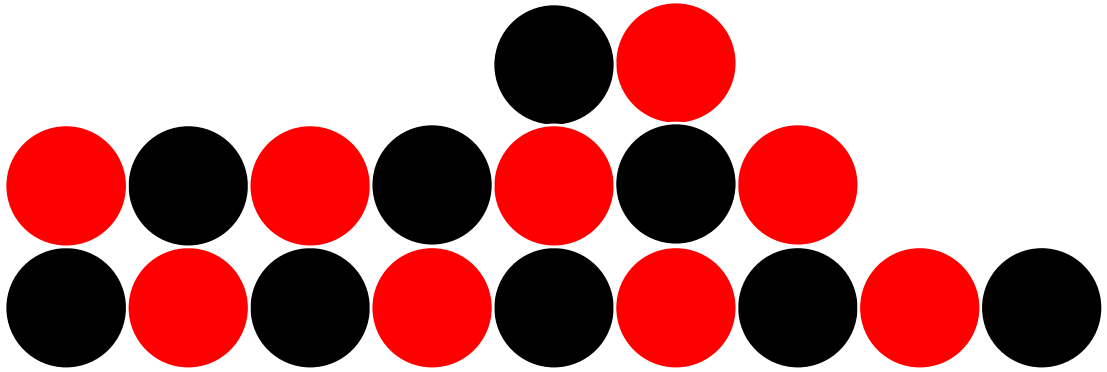
The foundation

Case 3 – chain of nodes



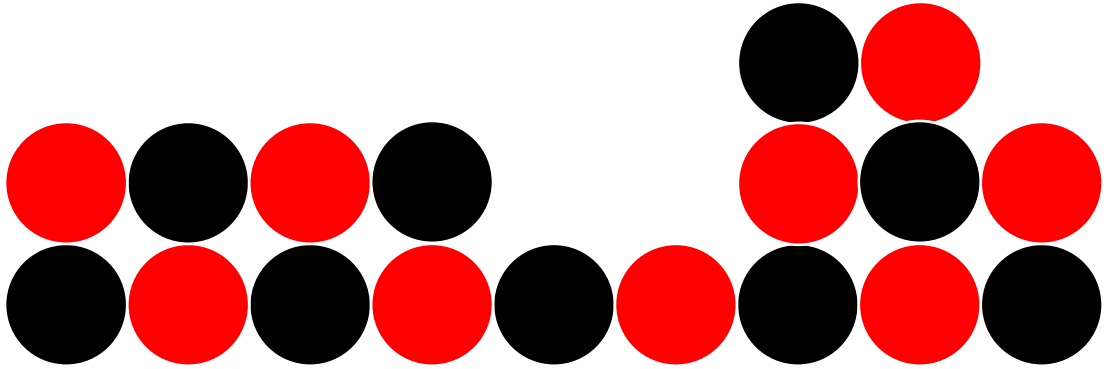
The foundation

Case 3 – chain of nodes



The foundation

Case 3 – chain of nodes



Summary and open problems

Minimal seed transformations:

Line to nice shape

Nice shape to nice shape

Open problems:

Decentralising the execution

Extending the class – shapes made of nice shapes?