https://hao-ai-lab.github.io/dsc204a-w24/

DSC 204A: Scalable Data Systems
Winter 2024

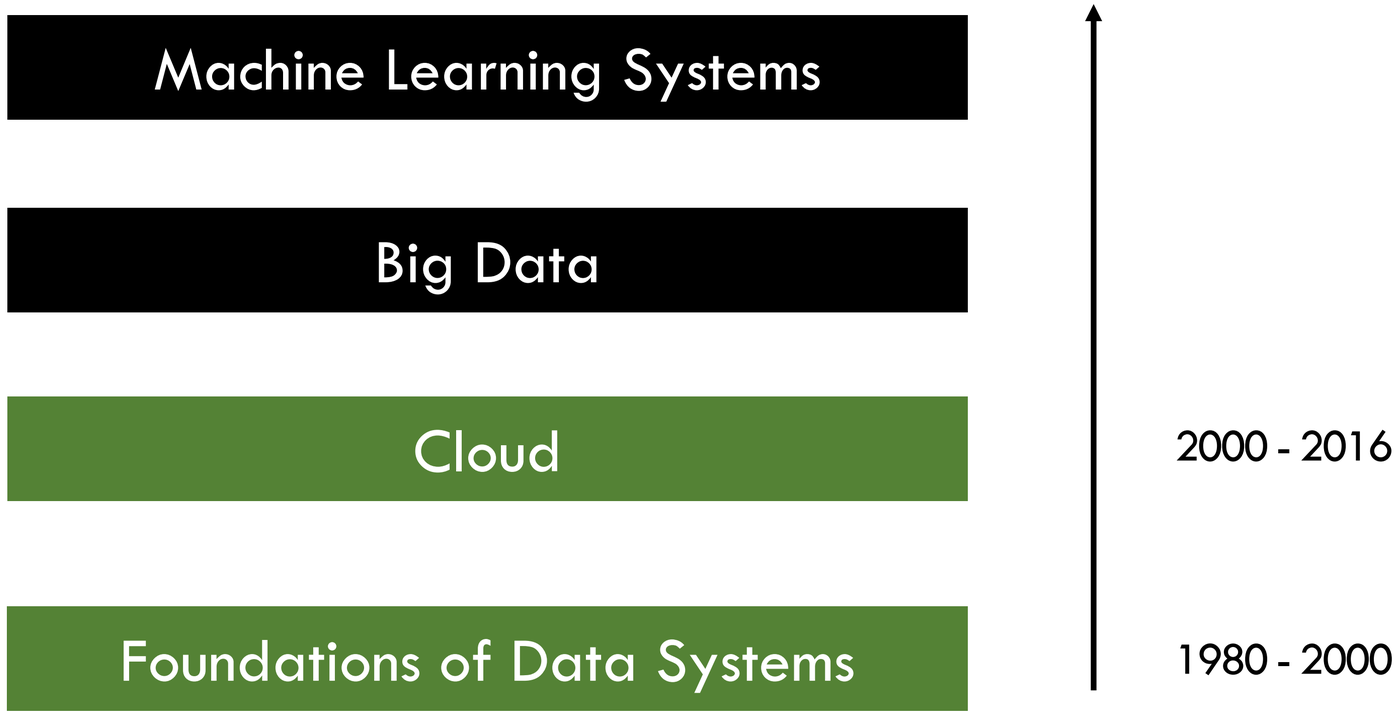
Machine Learning Systems

Big Data

Cloud

Foundations of Data Systems

Where We Are


Machine Learning Systems
Big Data
Cloud
Foundations of Data Systems
2000 - 2016
1980 - 2000

Today's topic

- Collective communication
  - Connection between distributed SGD and collective comm
  - Communication Model: $\alpha + n\,\beta, \beta = \frac{1}{B}$
  - Small Message size ($n \to 0$): α dominates, emphasize latency
  - Large Message Size ($n \to +\infty$): $n\,\beta$ dominate, emphasize bandwidth utilization

Recap: Minimum Spanning Tree Algorithm

Reduce(-to-one)

$$\log(p)(\alpha + n\beta + n\gamma)$$

Scatter

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Gather

$$\log(p)\alpha + \frac{p-1}{p} n\beta$$

Broadcast

$$\log(p)(\alpha + n\beta)$$

Reduce-scatter

$$2\log(p)\alpha + \log(p)n(\beta + \gamma) + \frac{p-1}{p} n\beta$$

Allreduce

$$2\log(p)\alpha + \log(p)n(2\beta + \gamma)$$

Allgather

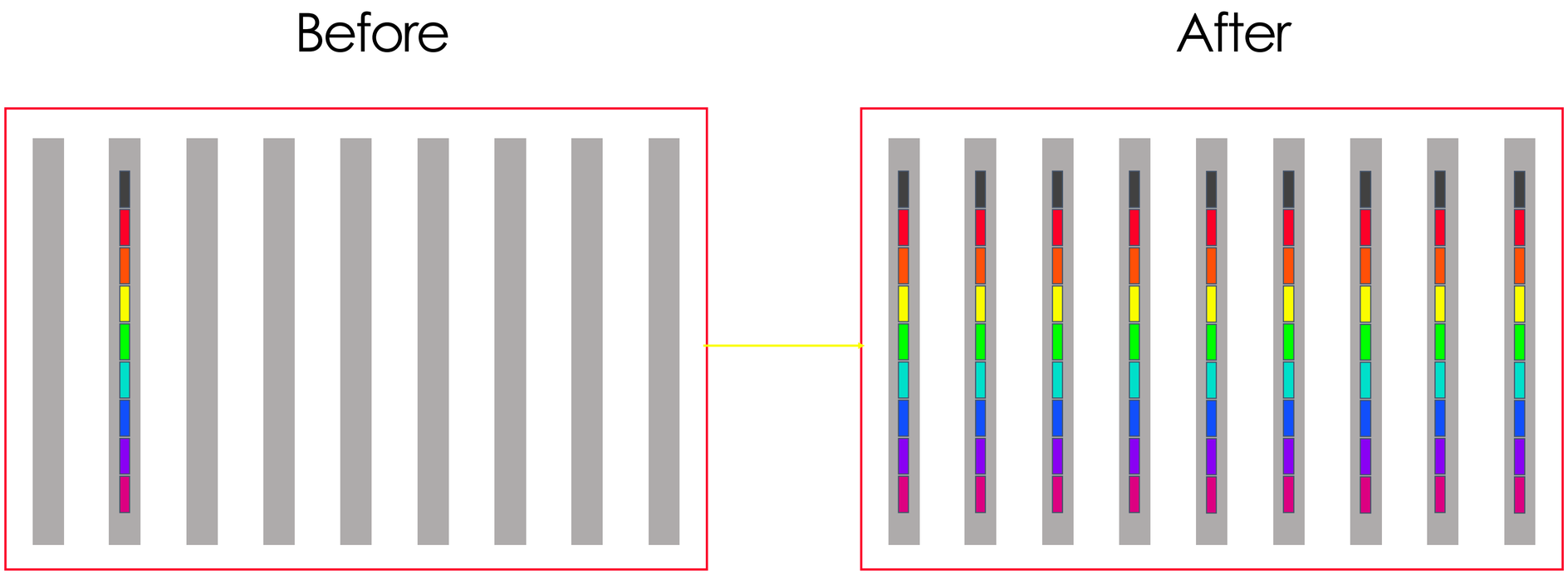
$$2\log(p)\alpha + \log(p)n\beta + \frac{p-1}{p} n\beta$$

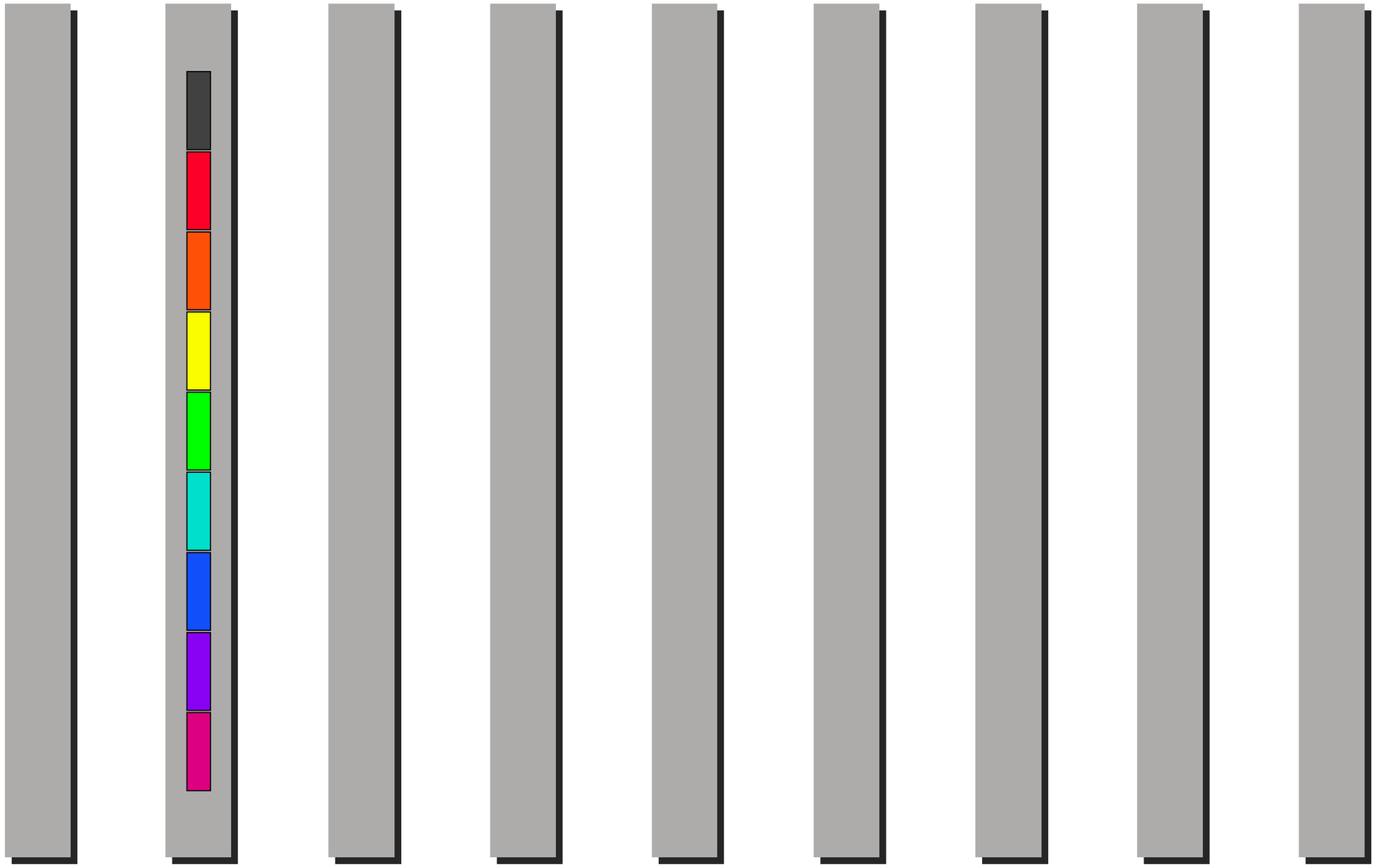
Pros and Cons of MST algorithms

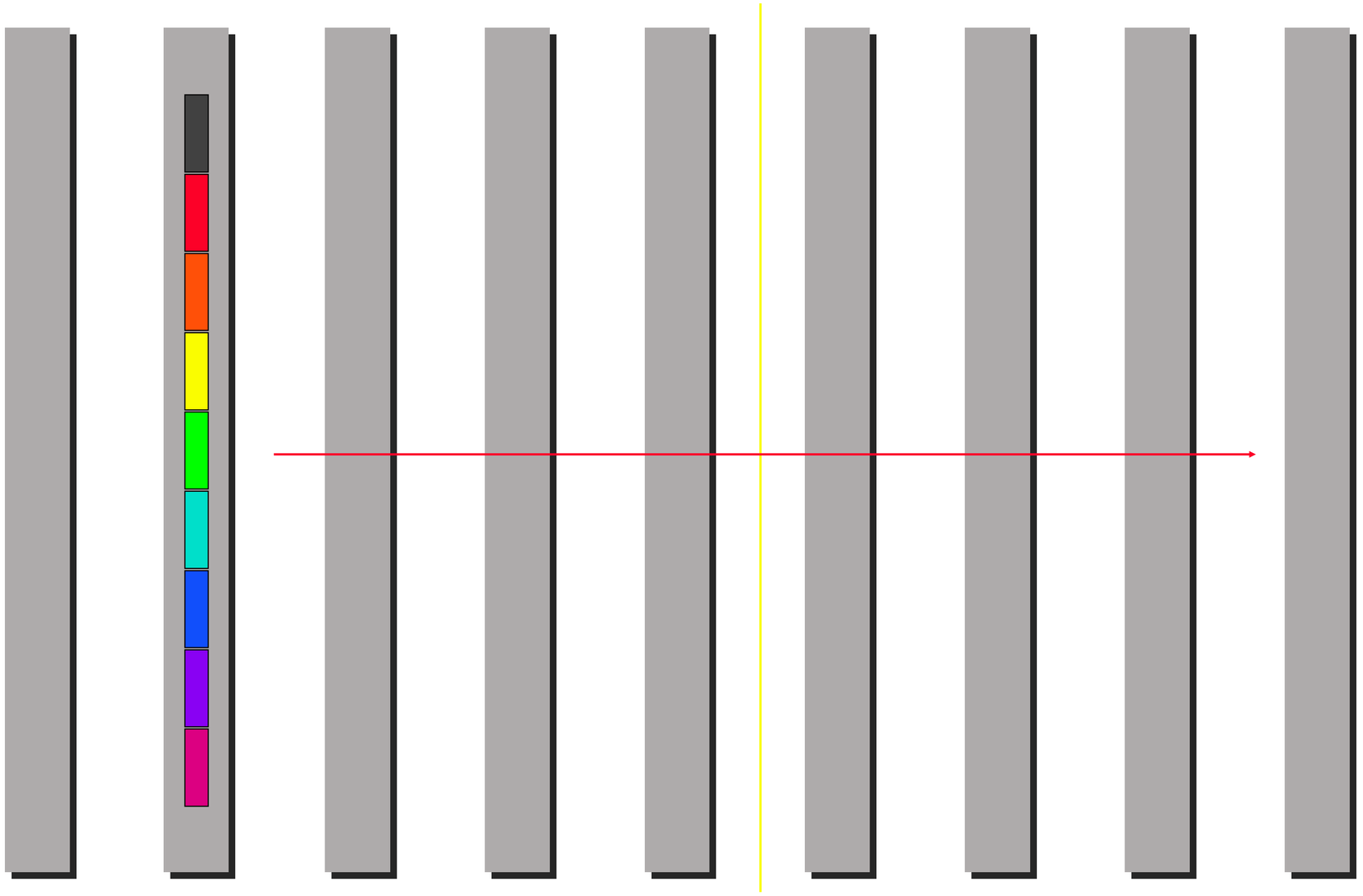
- Emphasize **low latency**
  - MST-based algorithm is latency-optimal
  - **How to prove?** (Taking broadcast as an example)

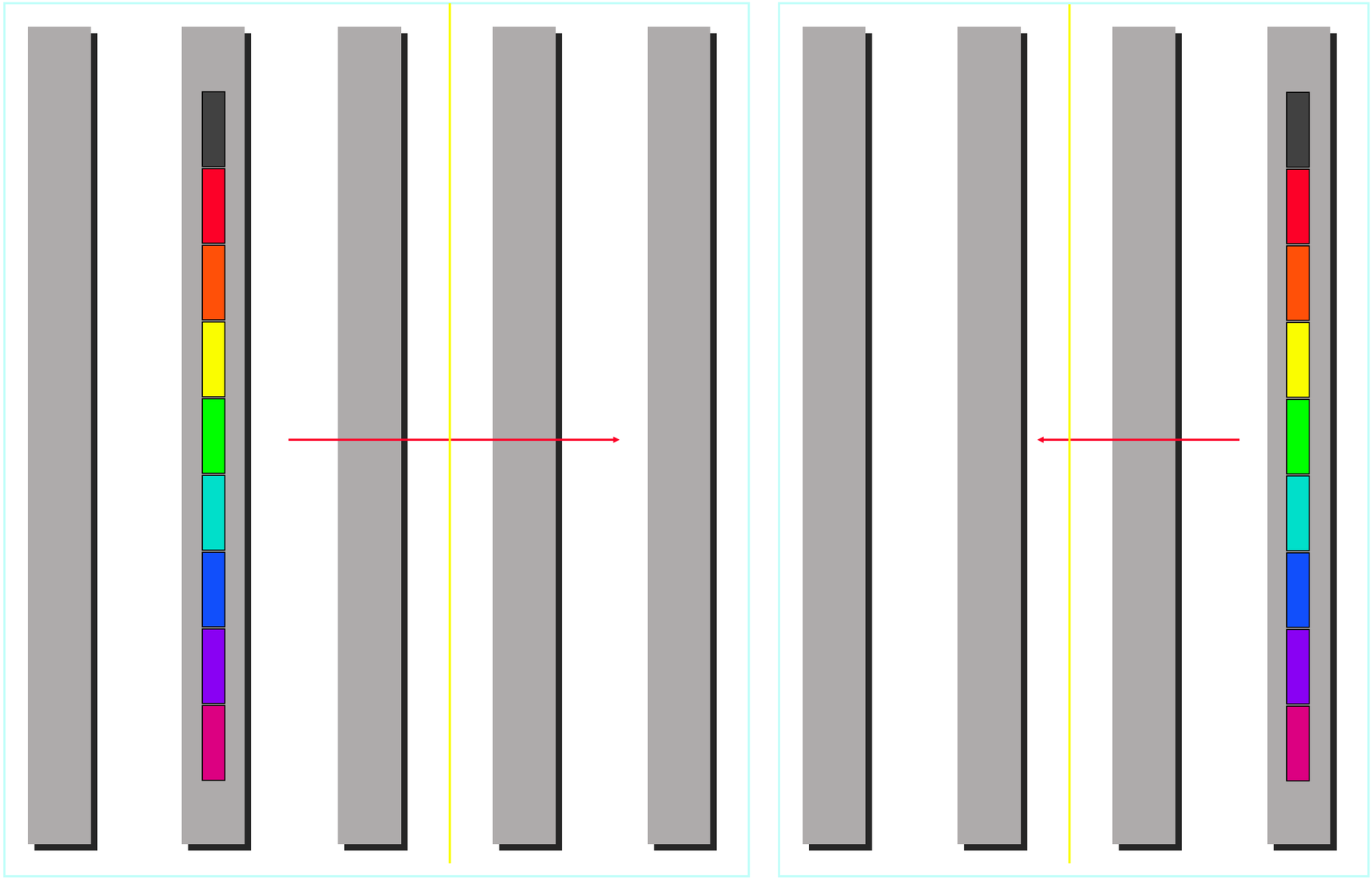
- Problem of Minimum Spanning Tree Algorithm?
  - Some links are idle

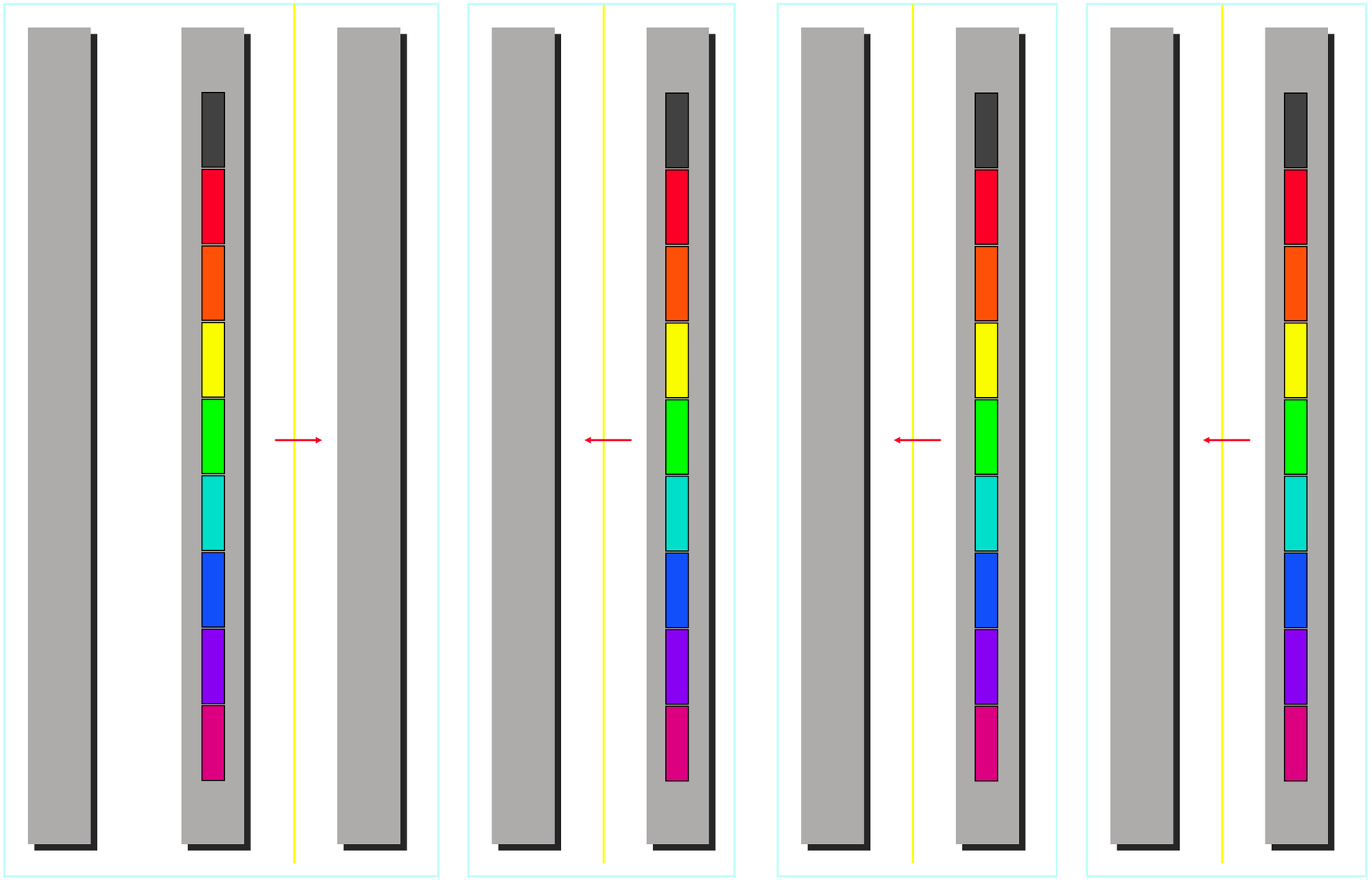
Broadcast

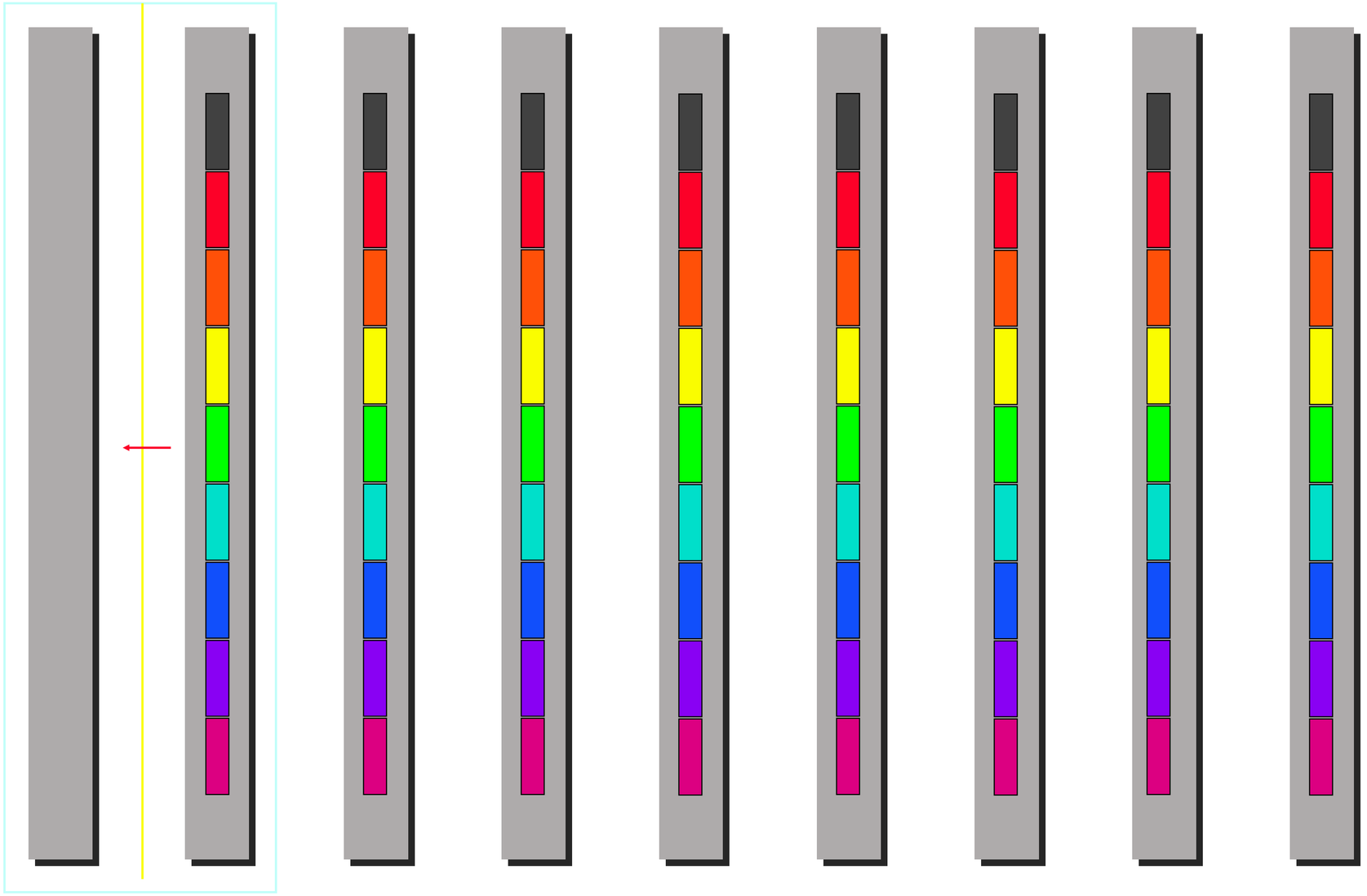


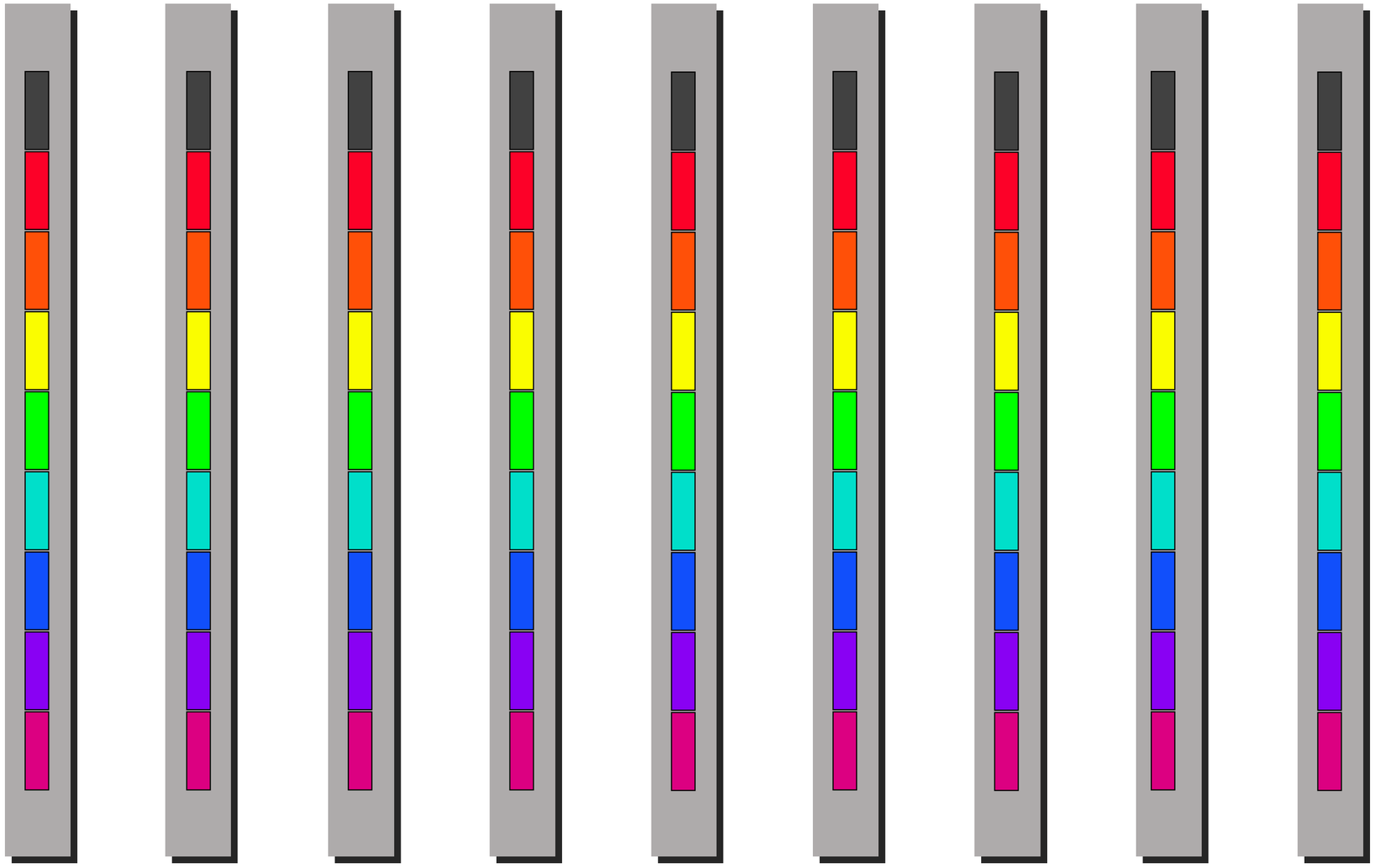












Large Message

Communication Model: $\alpha + n\,\beta, \beta = \frac{1}{B}$

- The second term dominates – we want to minimize the second term
  - We want to utilize the bandwidth as much as possible

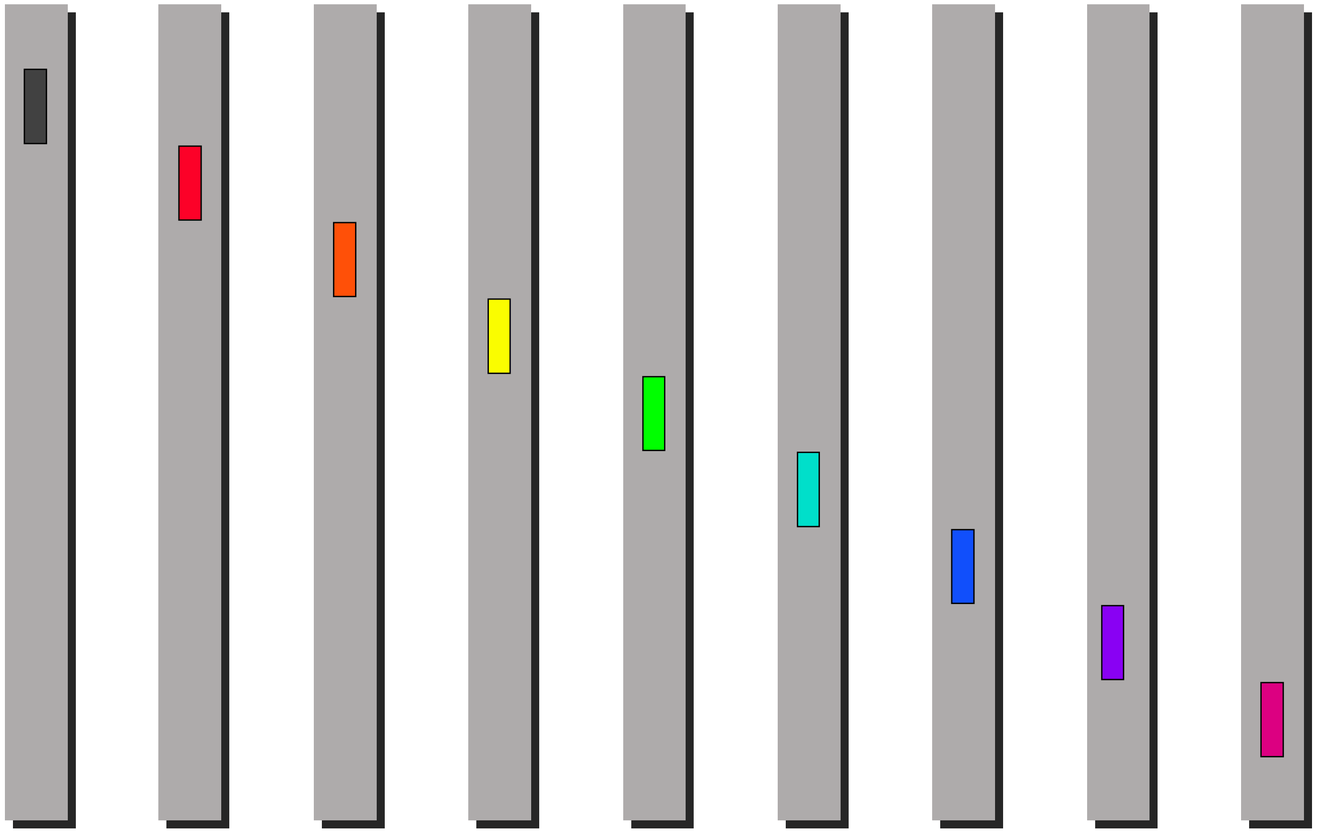
Long vector building blocks

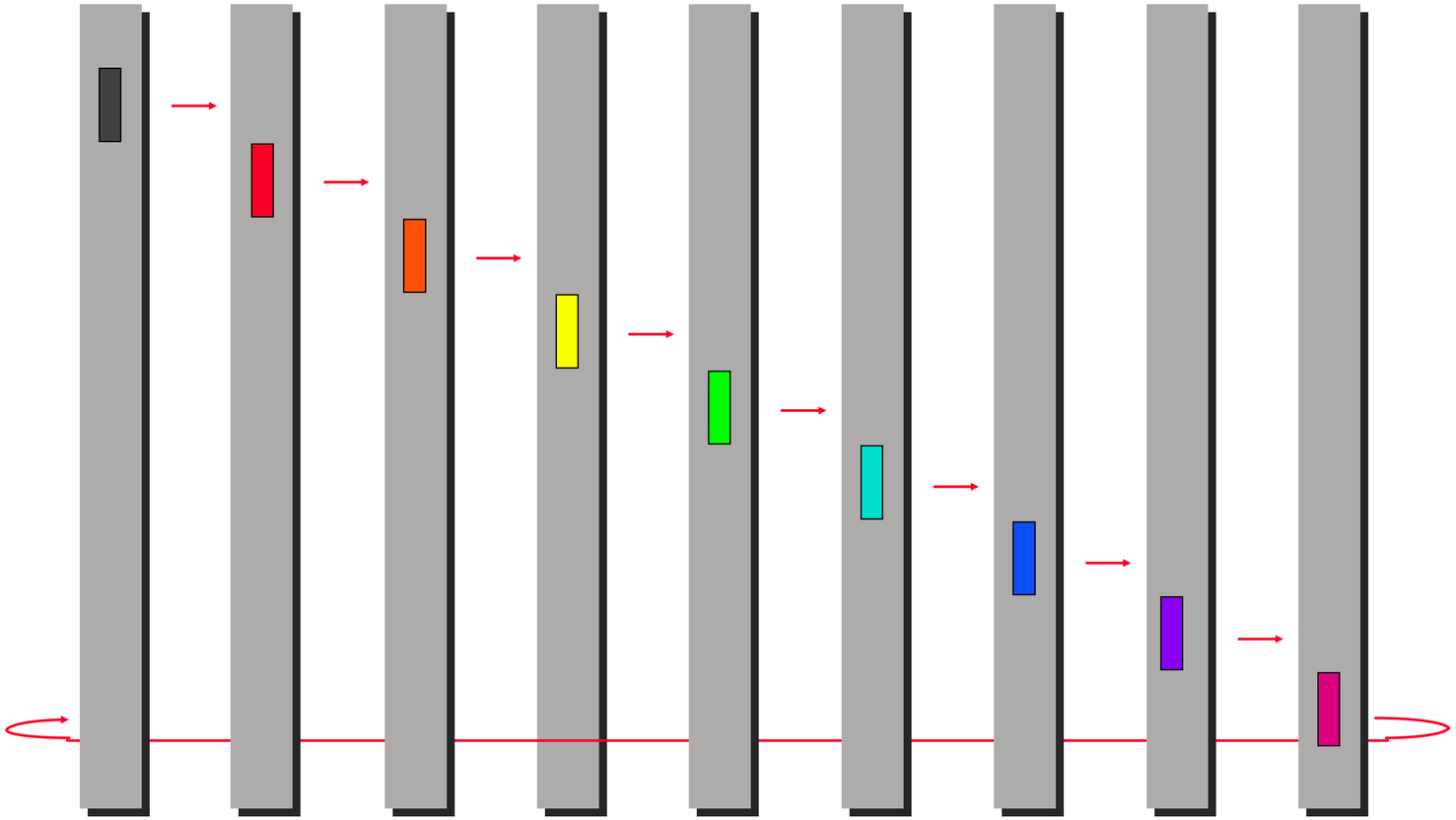
- We will show how the following building blocks:
  - collect/distributed combine
  - scatter/gather

  can be implemented using “bucket” algorithms while attaining
  - minimal cost due to length of vectors
  - implementation for arbitrary numbers of nodes
  - no network conflicts
- NOTICE: scatter and gather already satisfy these conditions

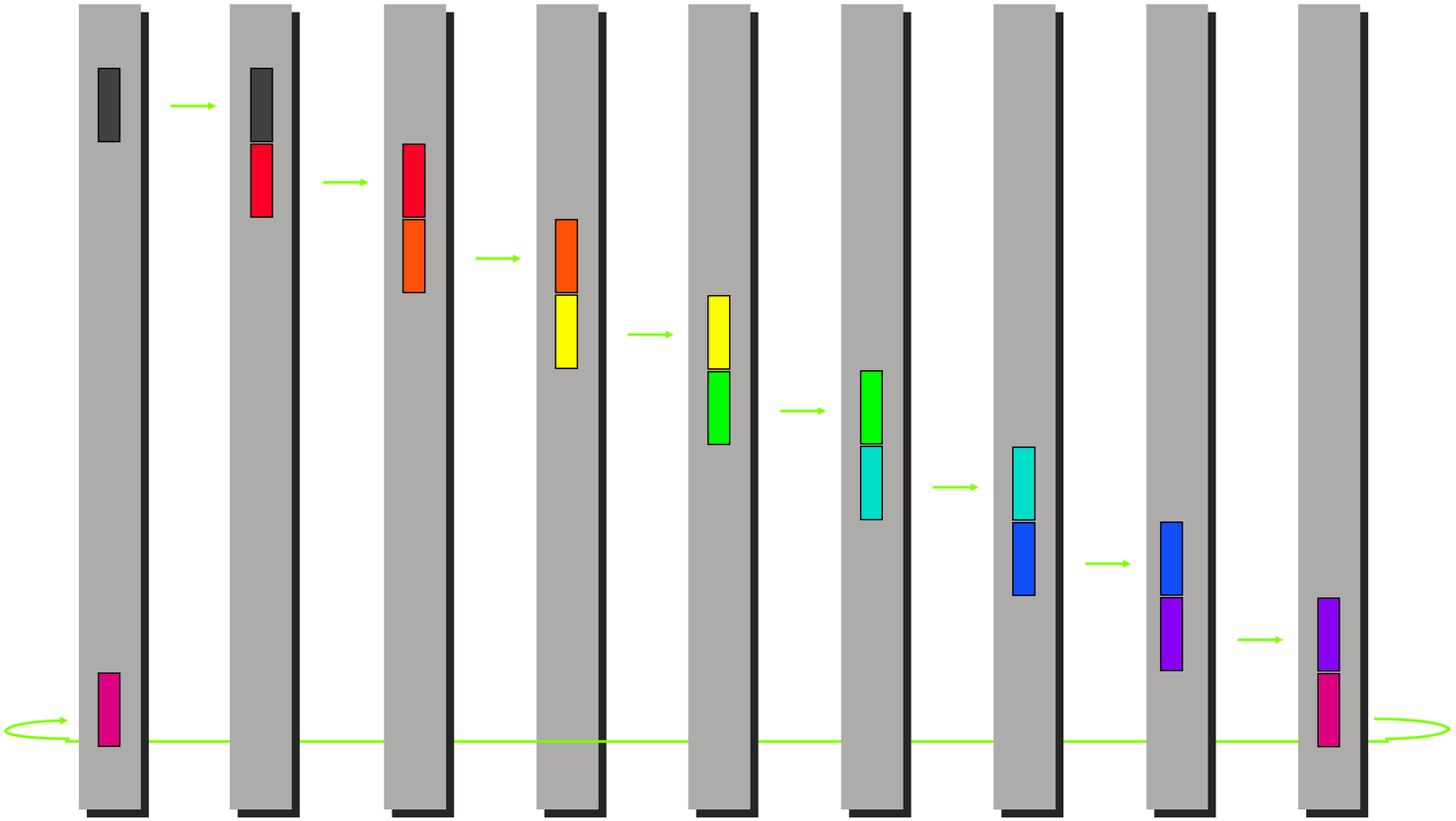
General principles

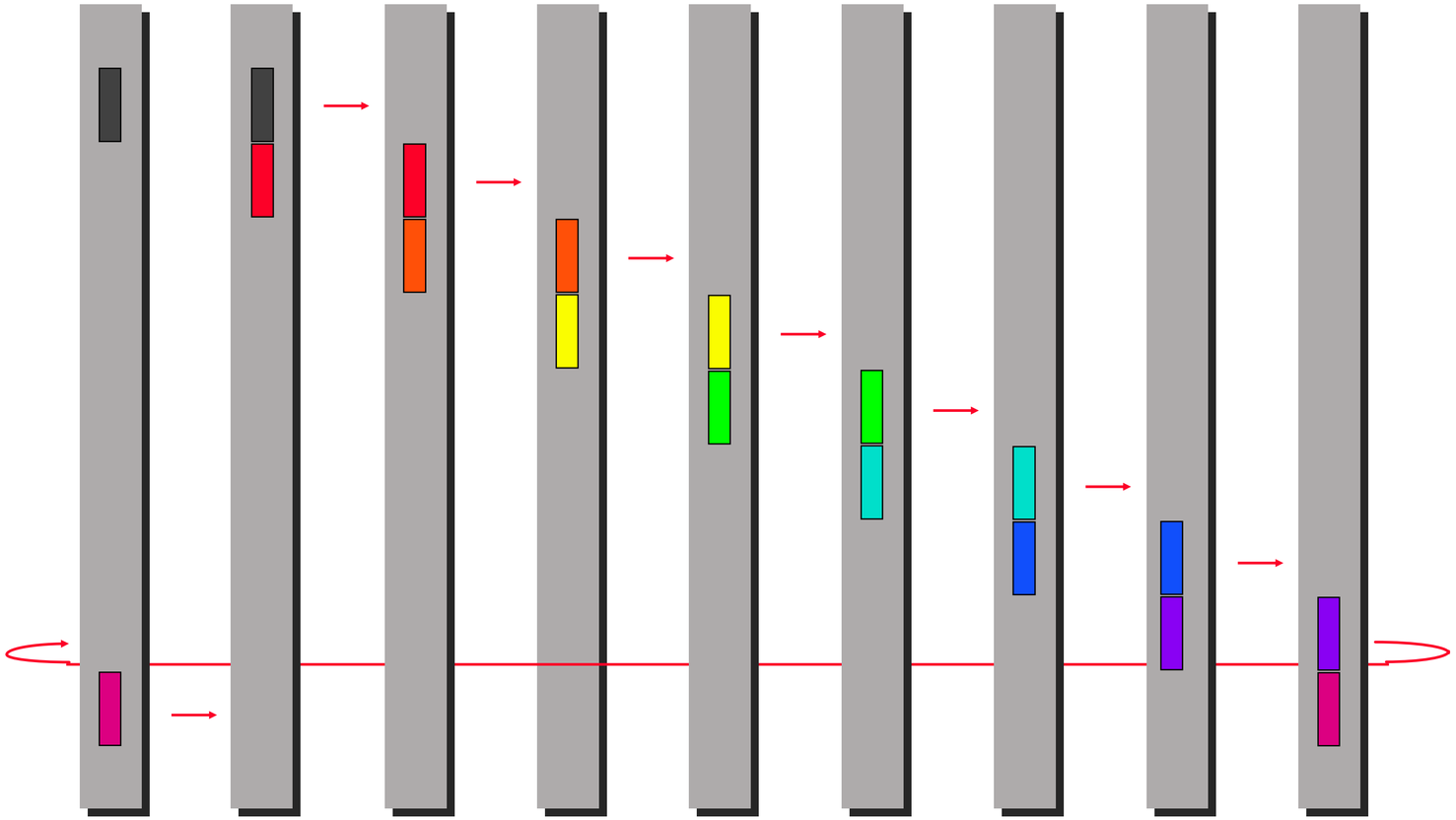
- Use all the links between every two nodes
- A logical ring can be embedded in a physical linear array with worm-hole routing, since the “wrap-around” message doesn’t conflict

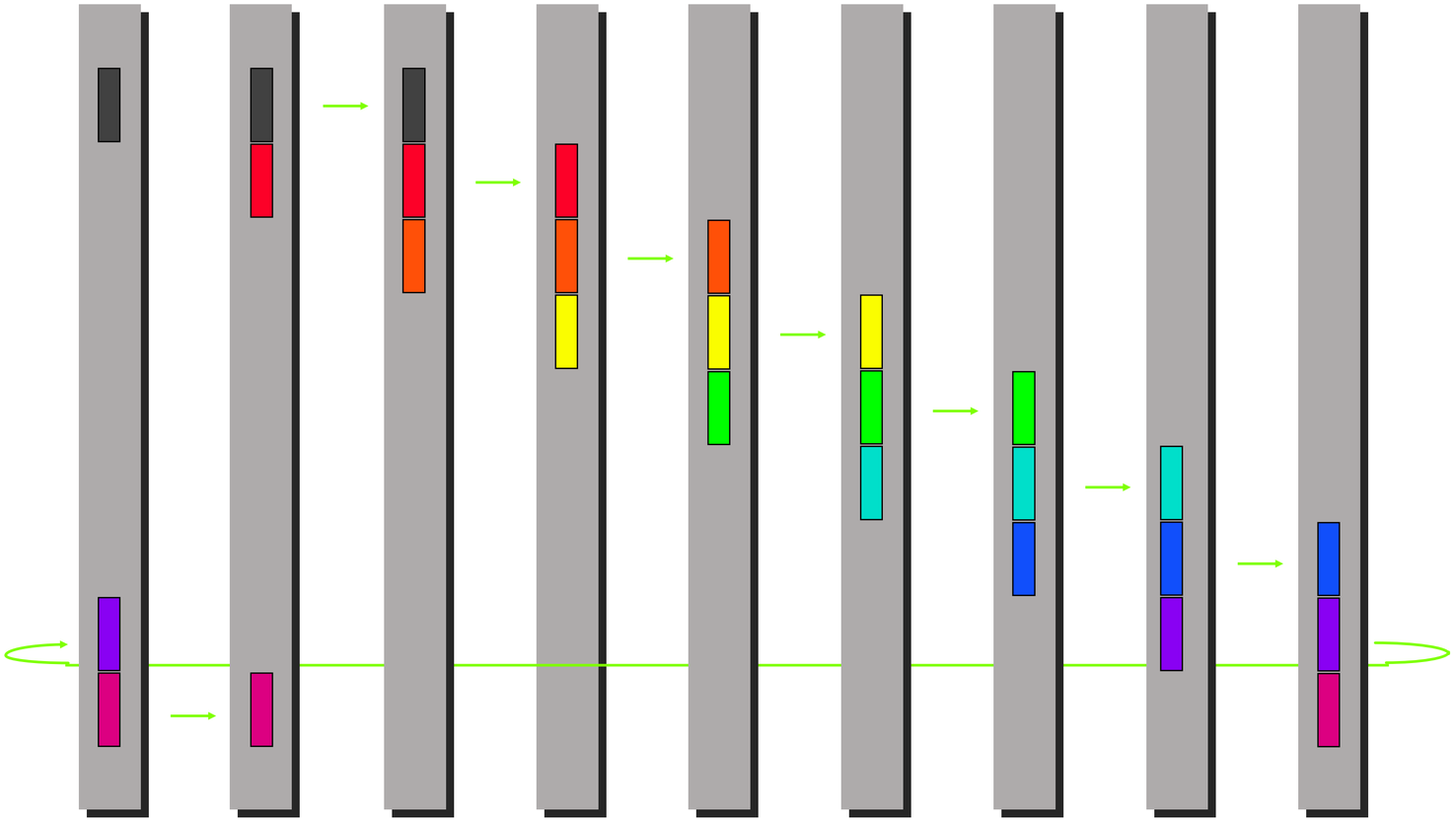




- A logical ring can be embedded in a physical linear array with worm-hole routing, since the “wrap-around” message doesn’t conflict





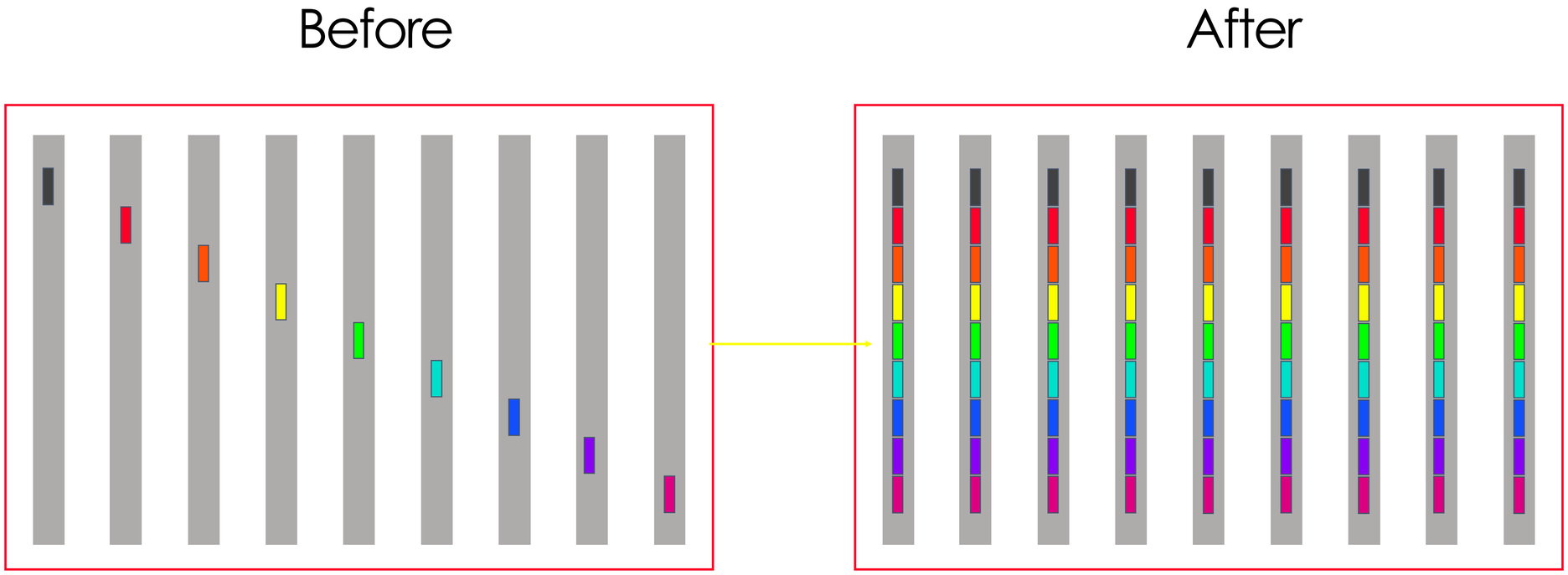


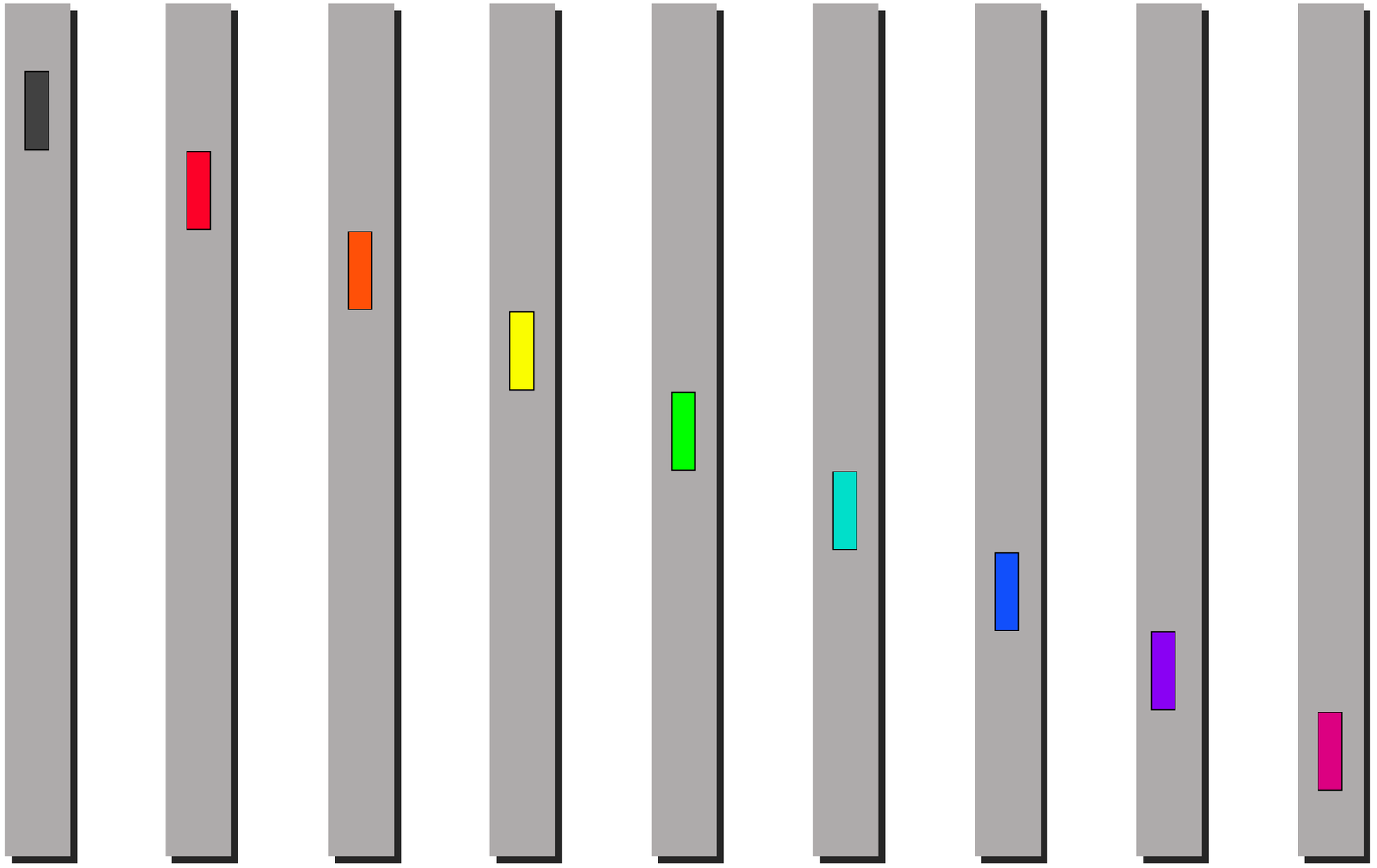
General principles

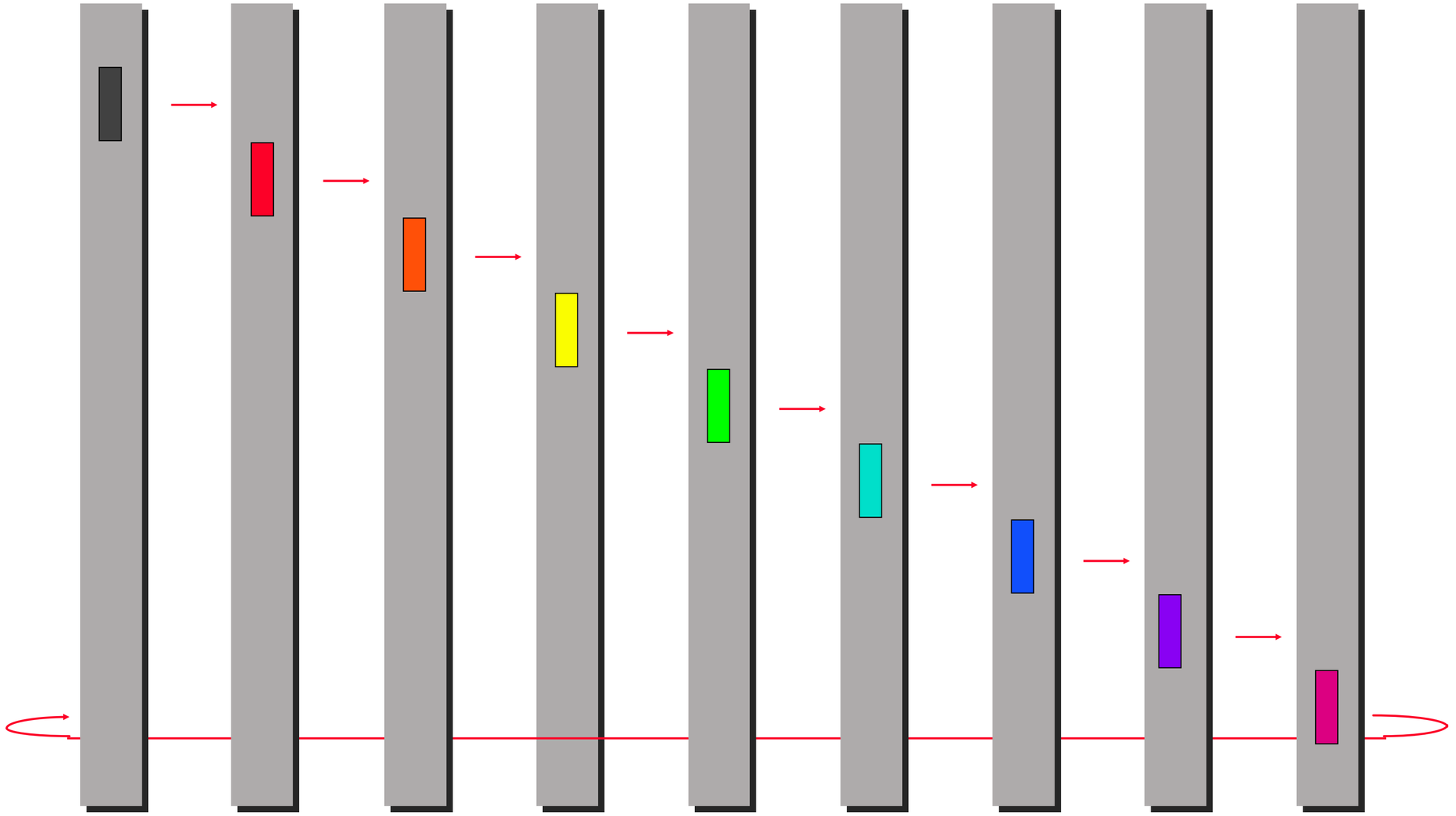
Ring algorithm has the following advantages

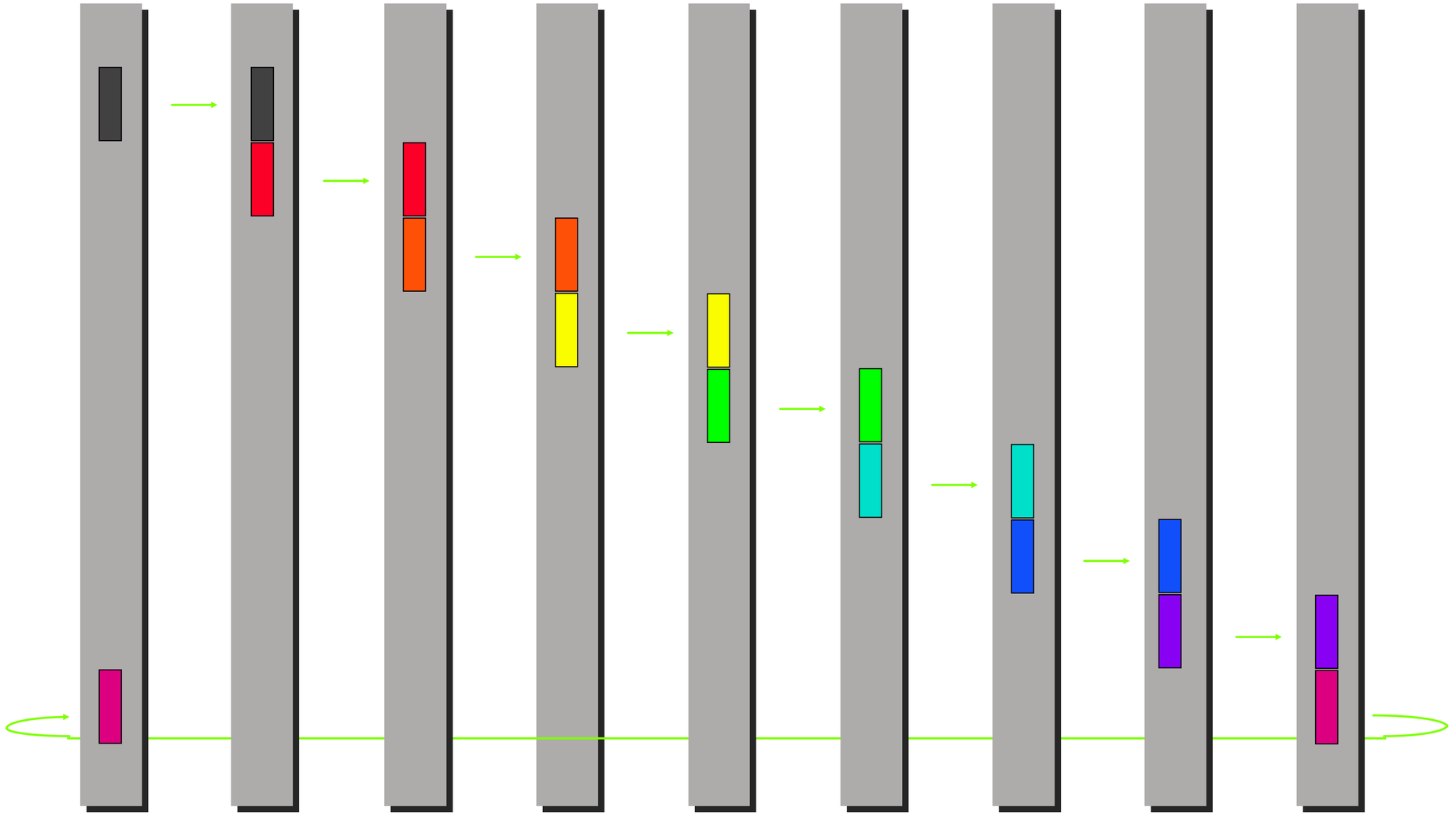
- Fully utilize the bandwidth (bandwidth optimal)
- implementation for arbitrary numbers of node

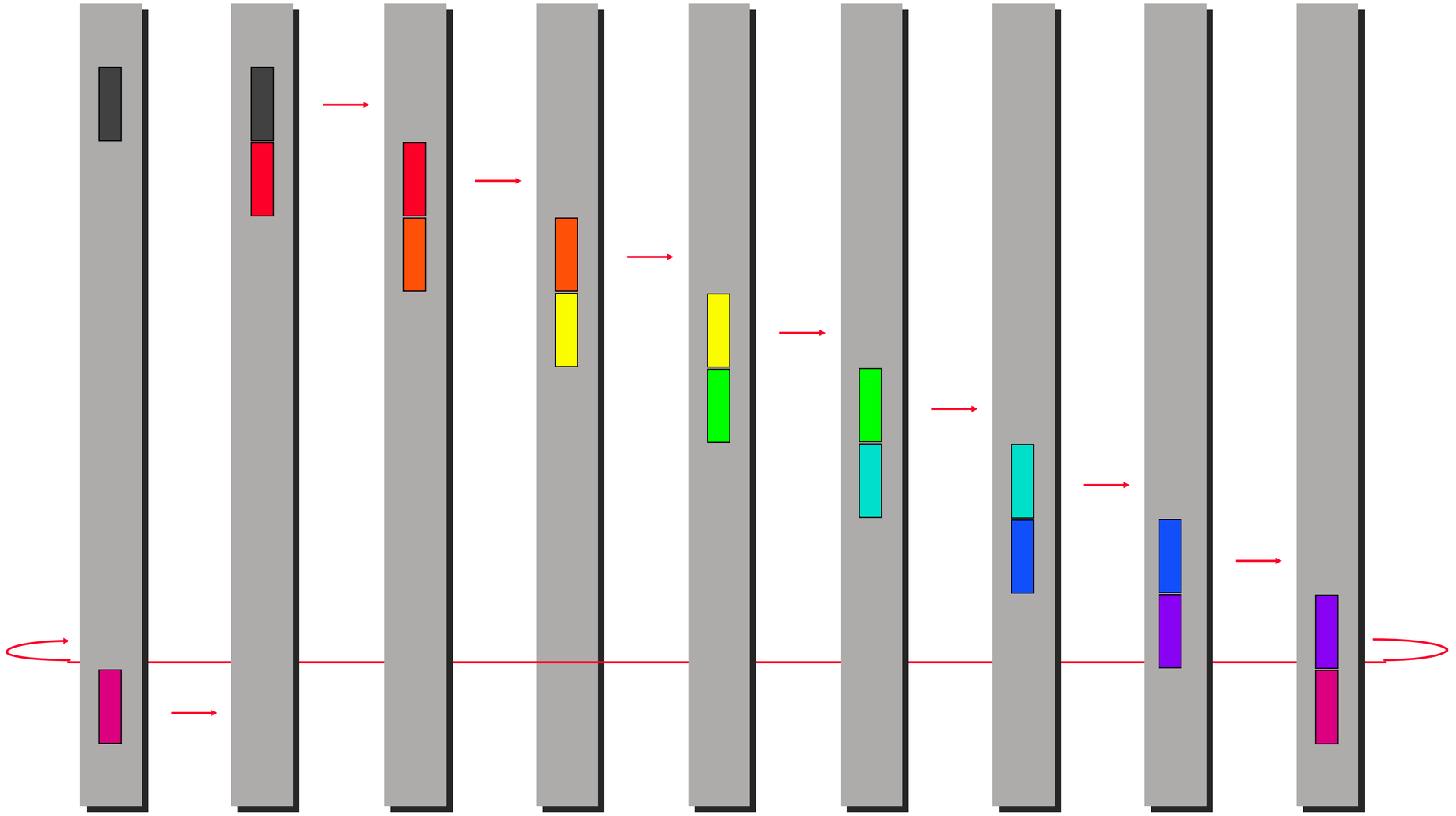
Allgather

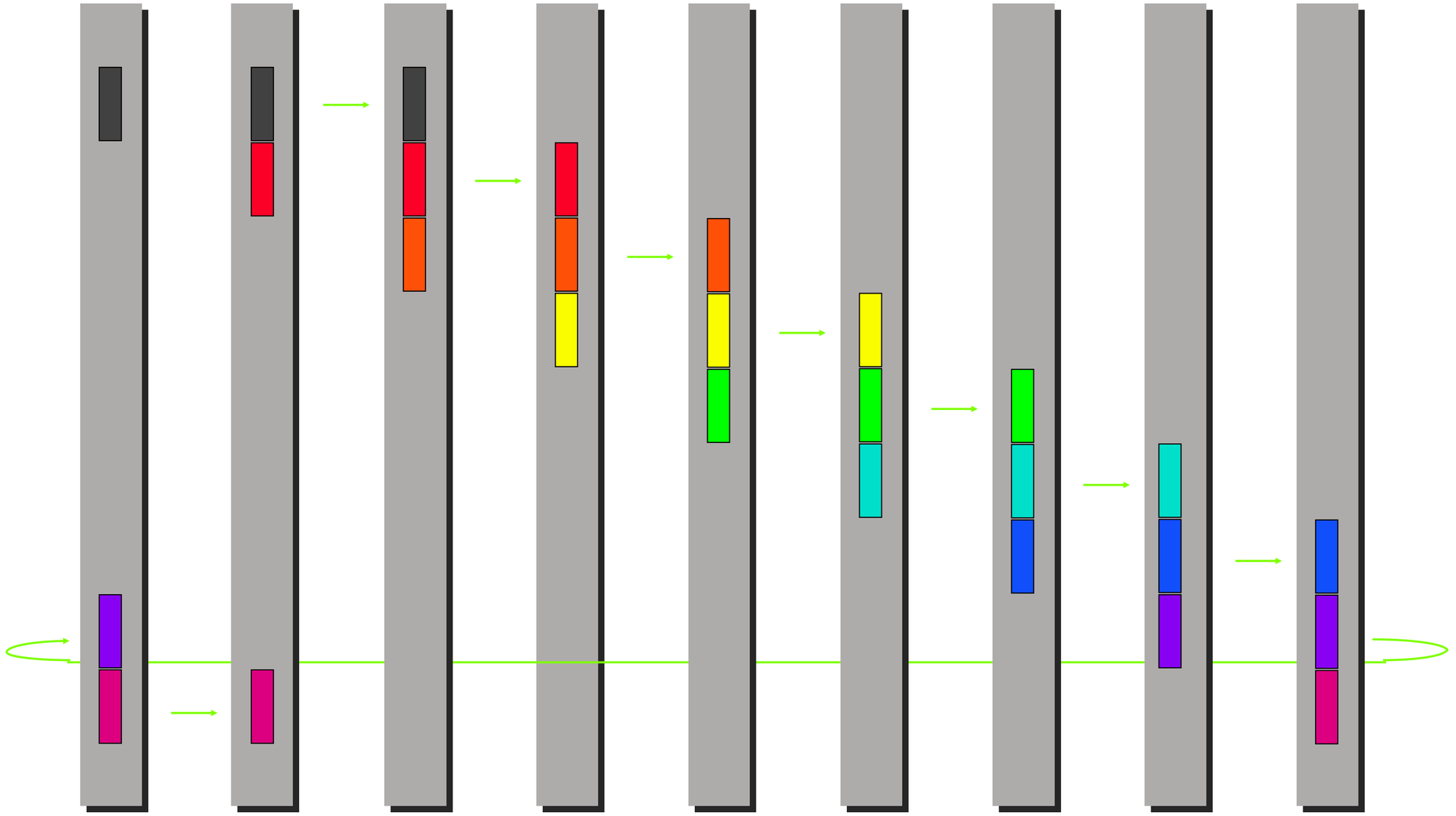


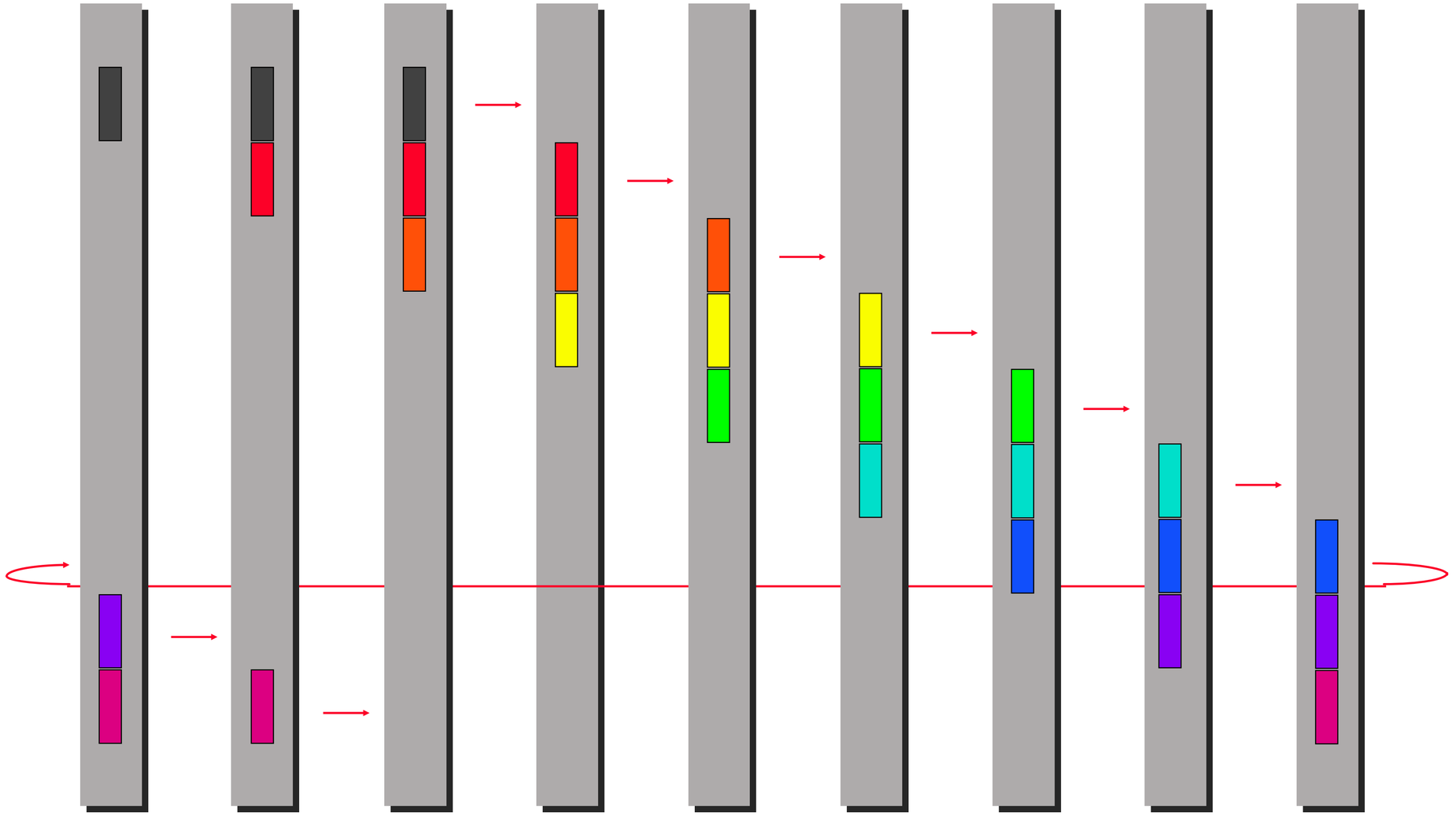


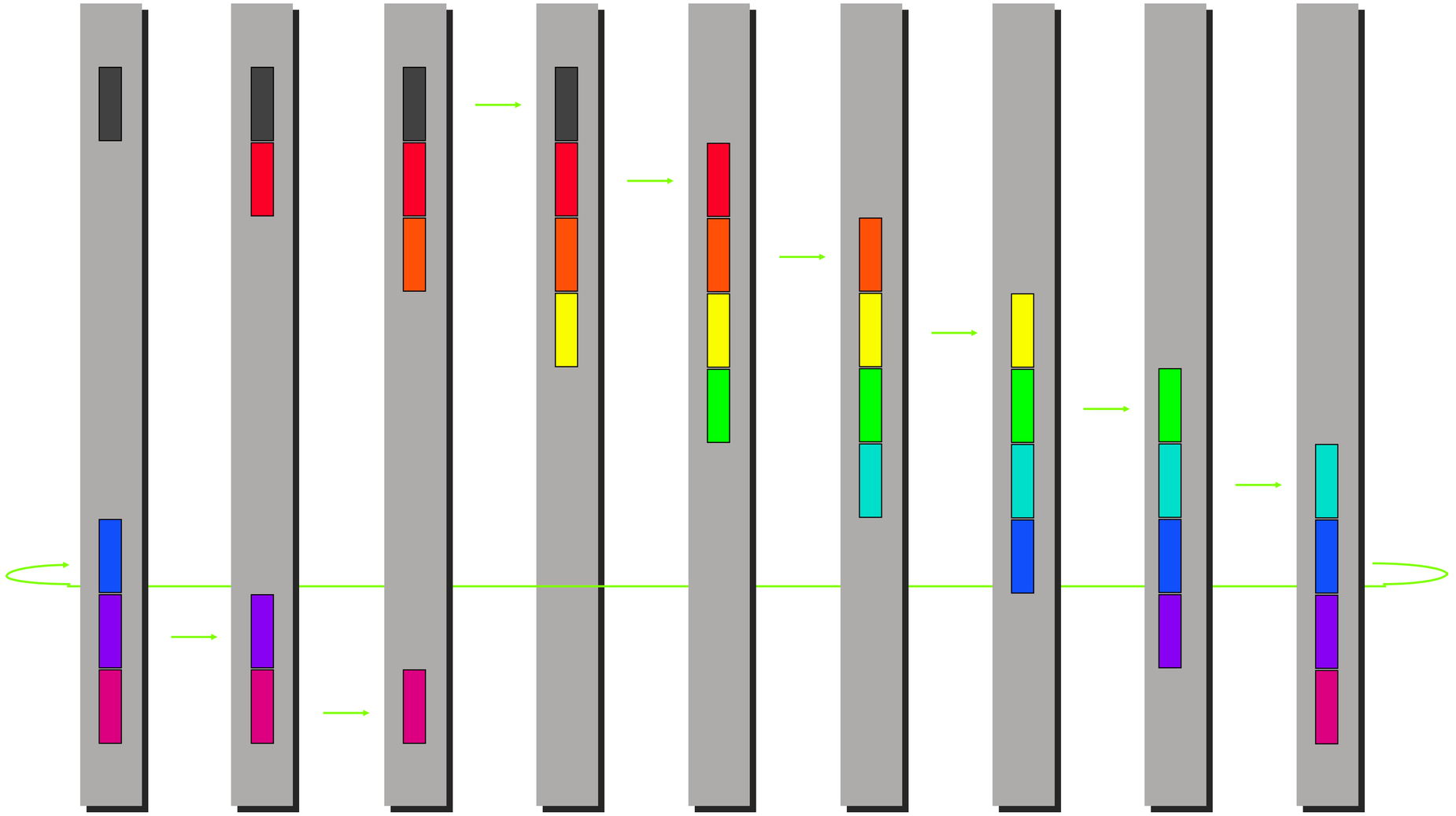


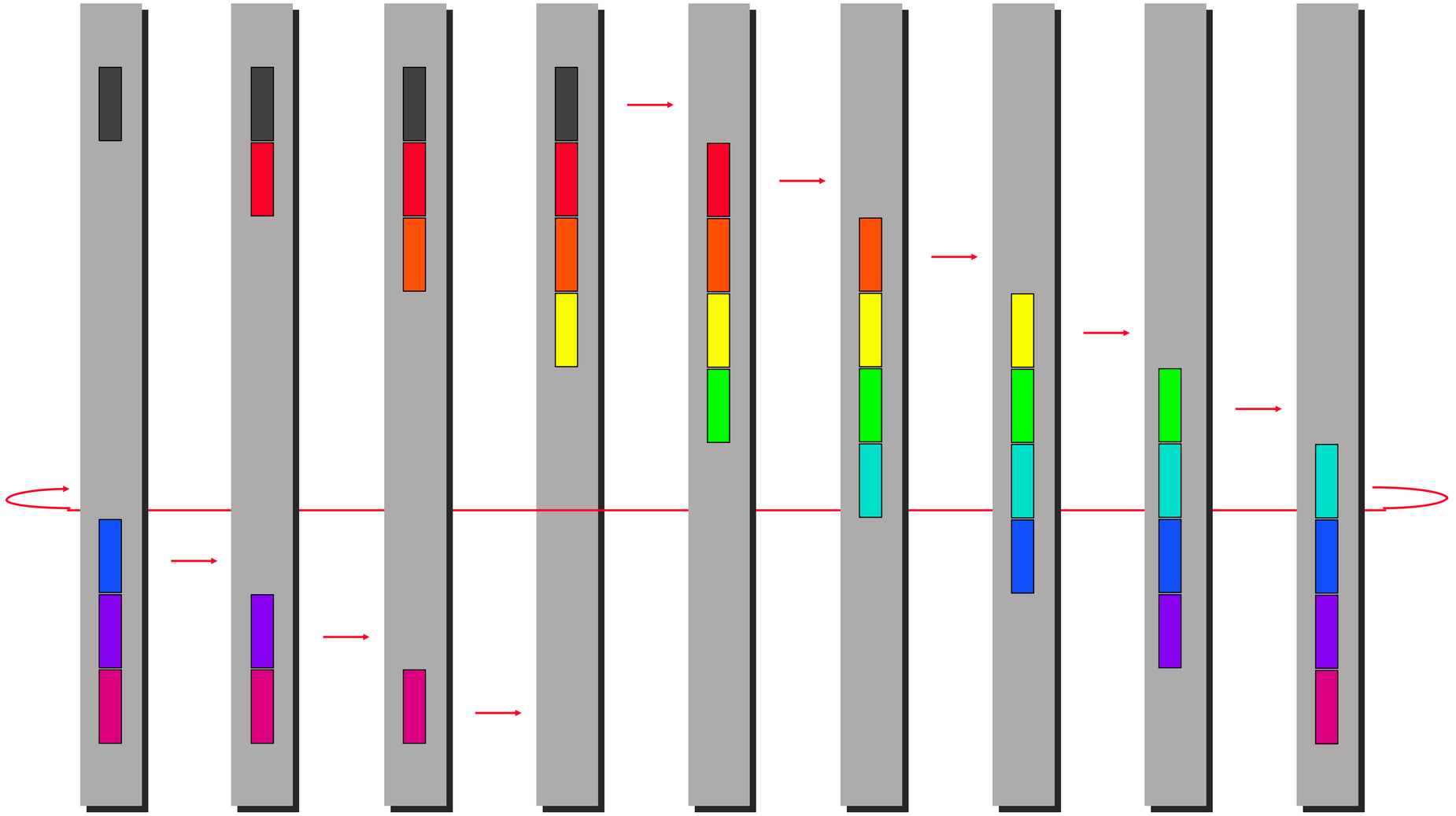


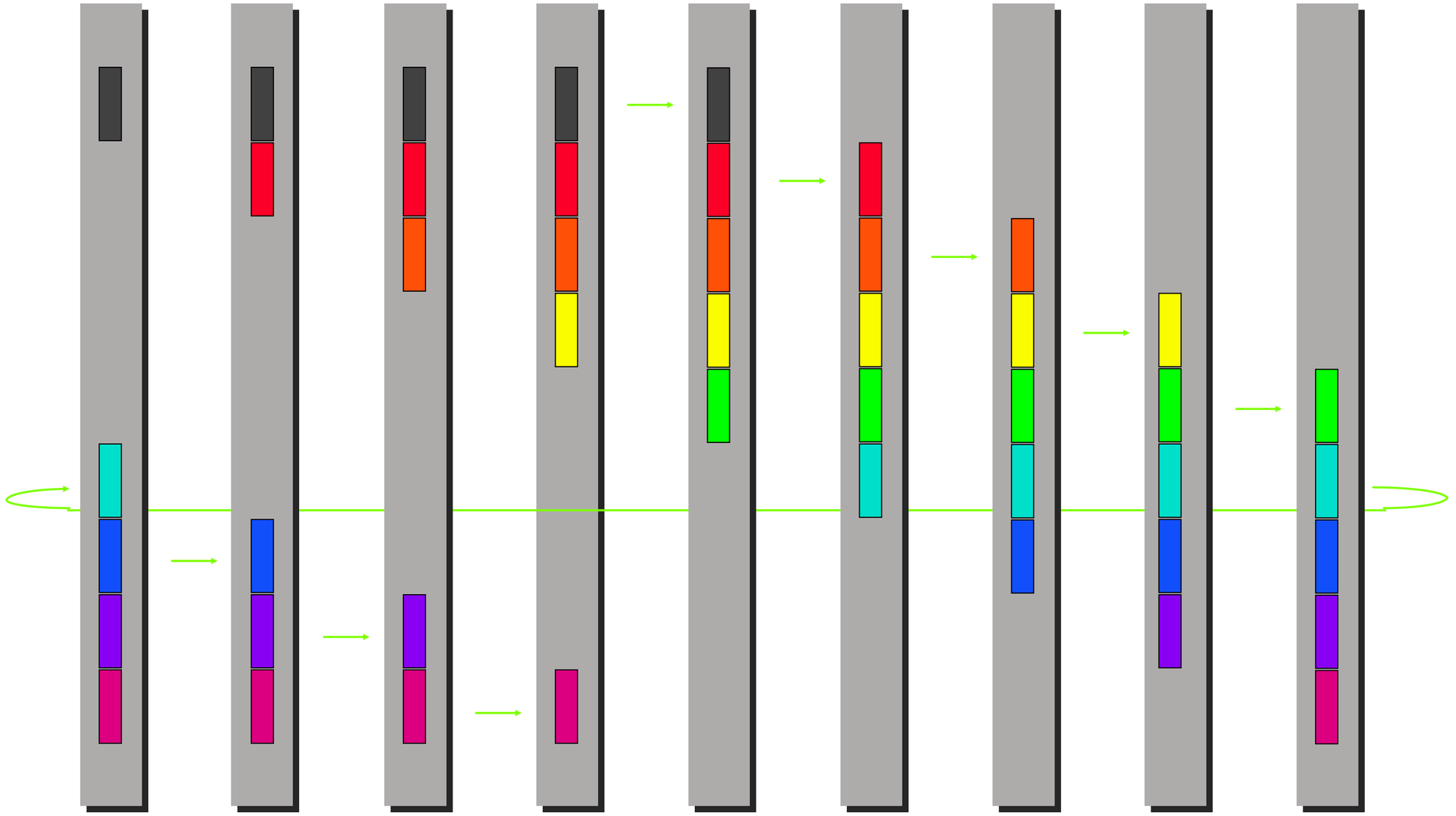


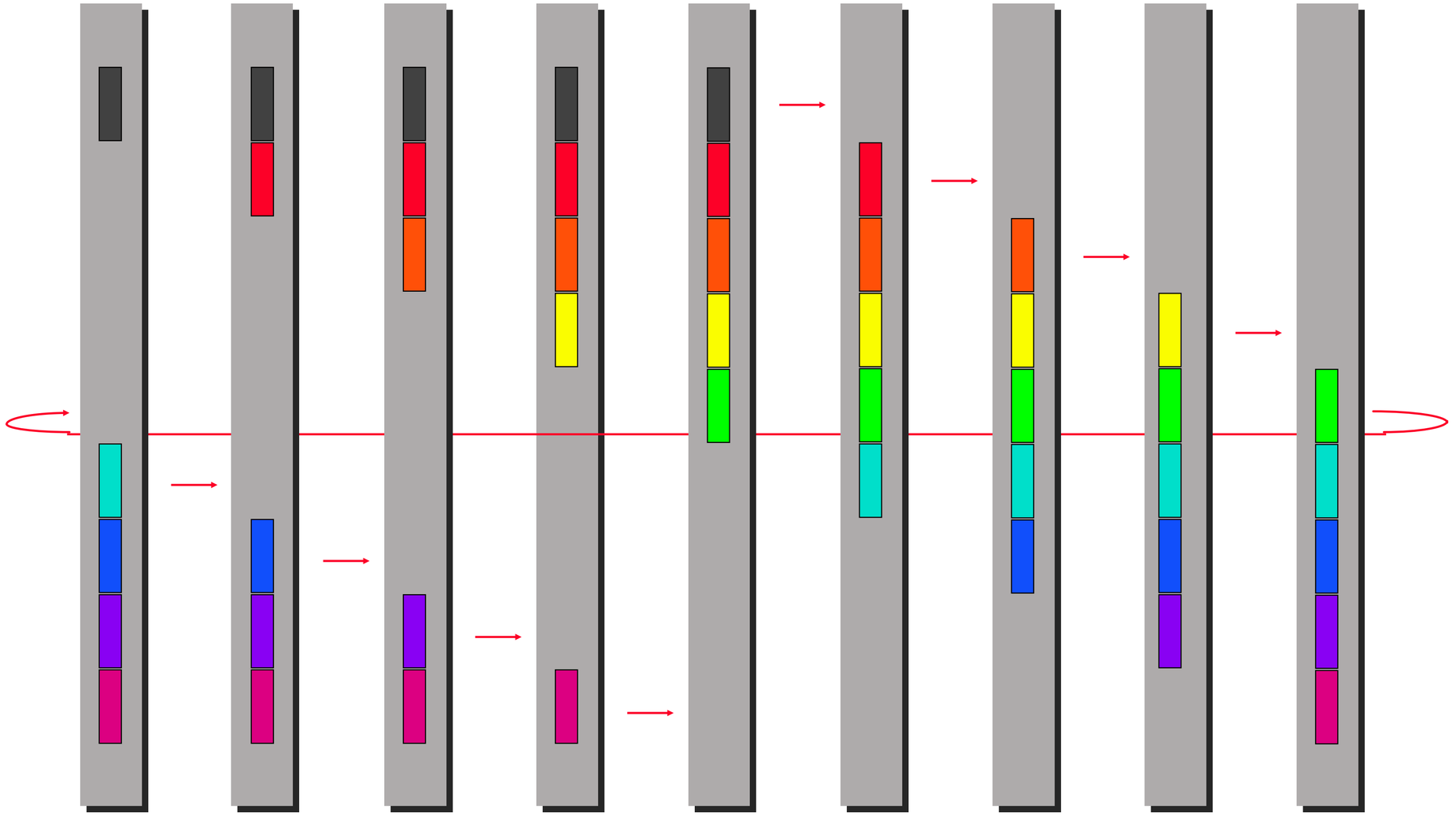


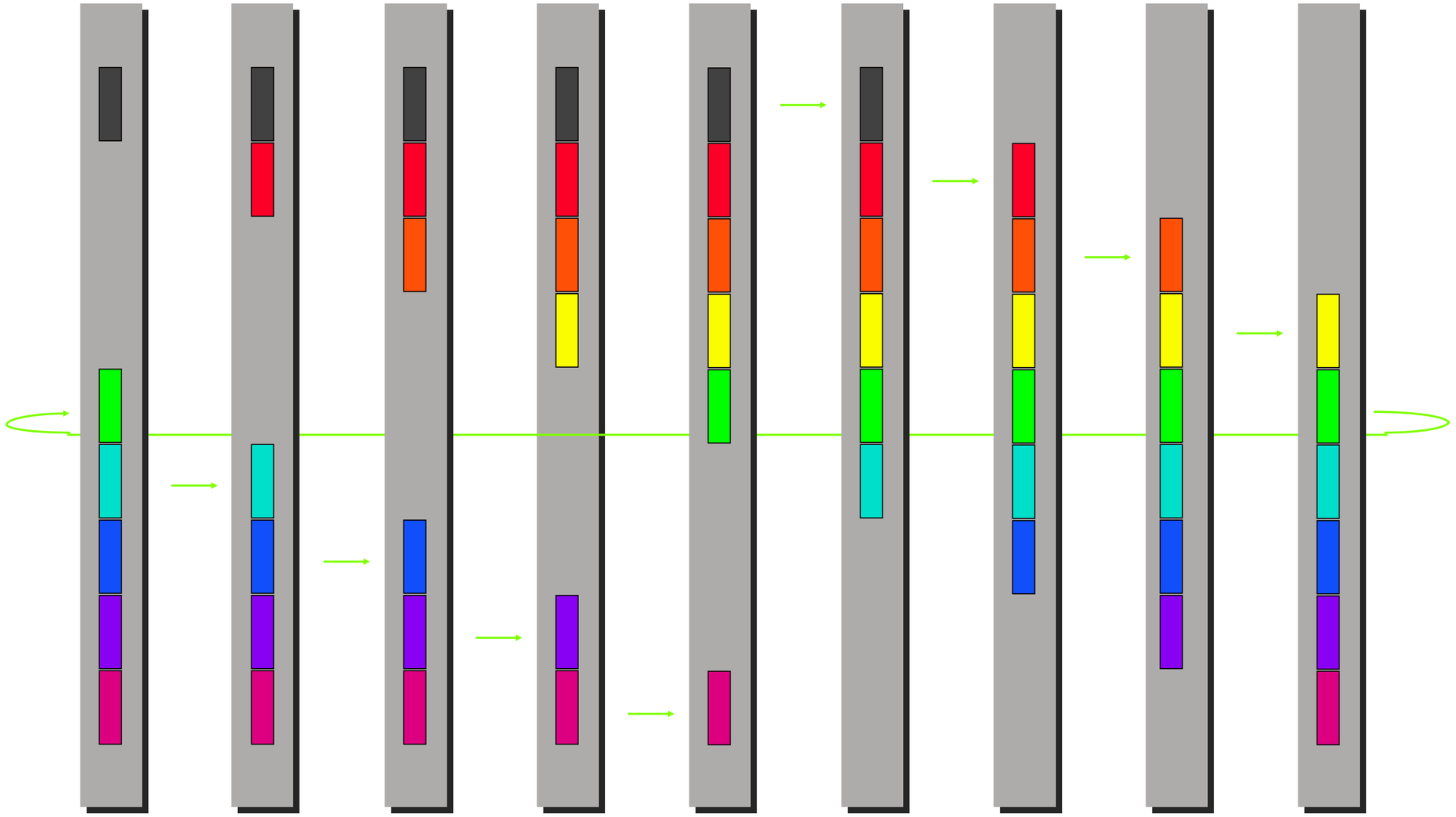


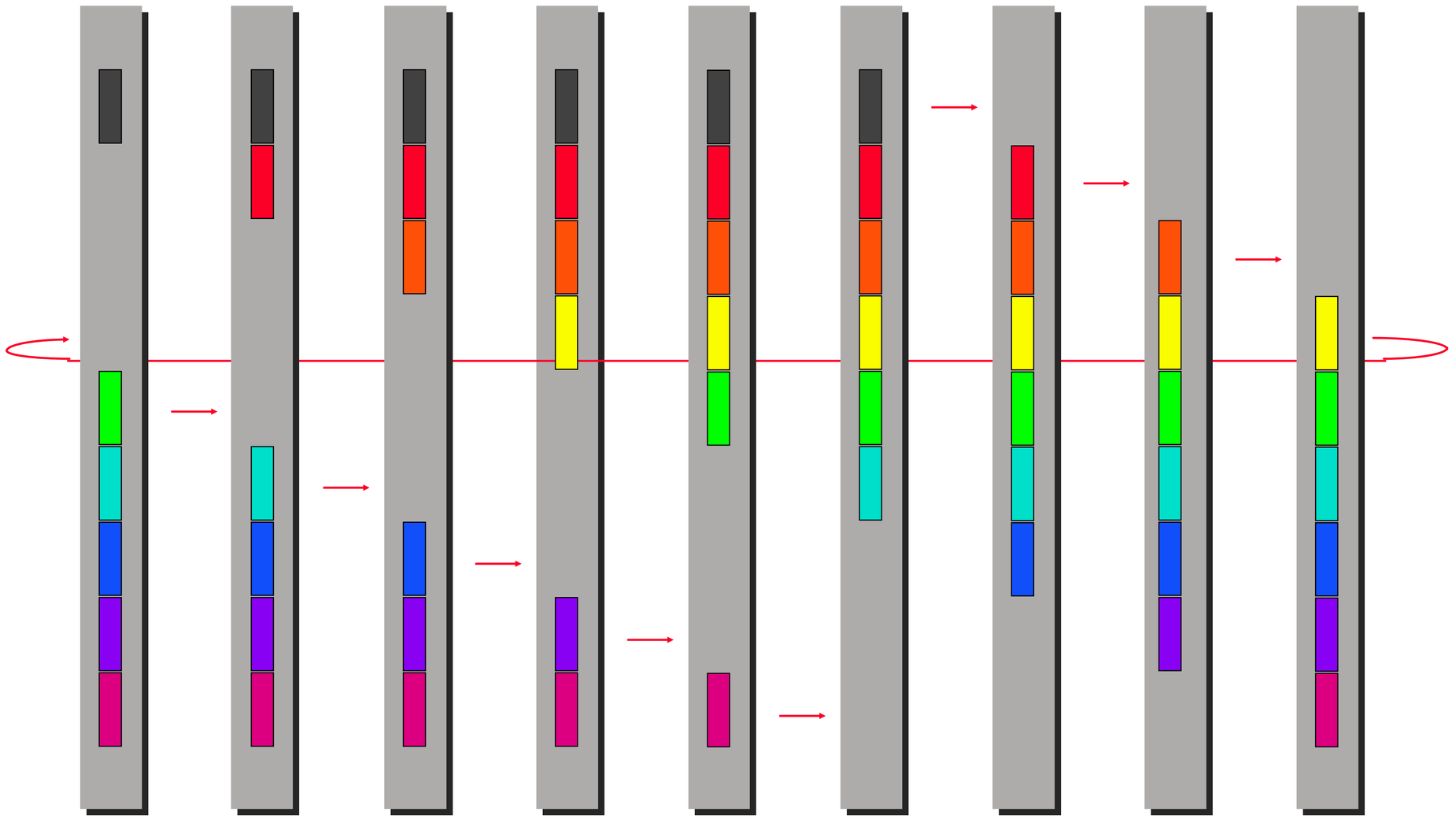


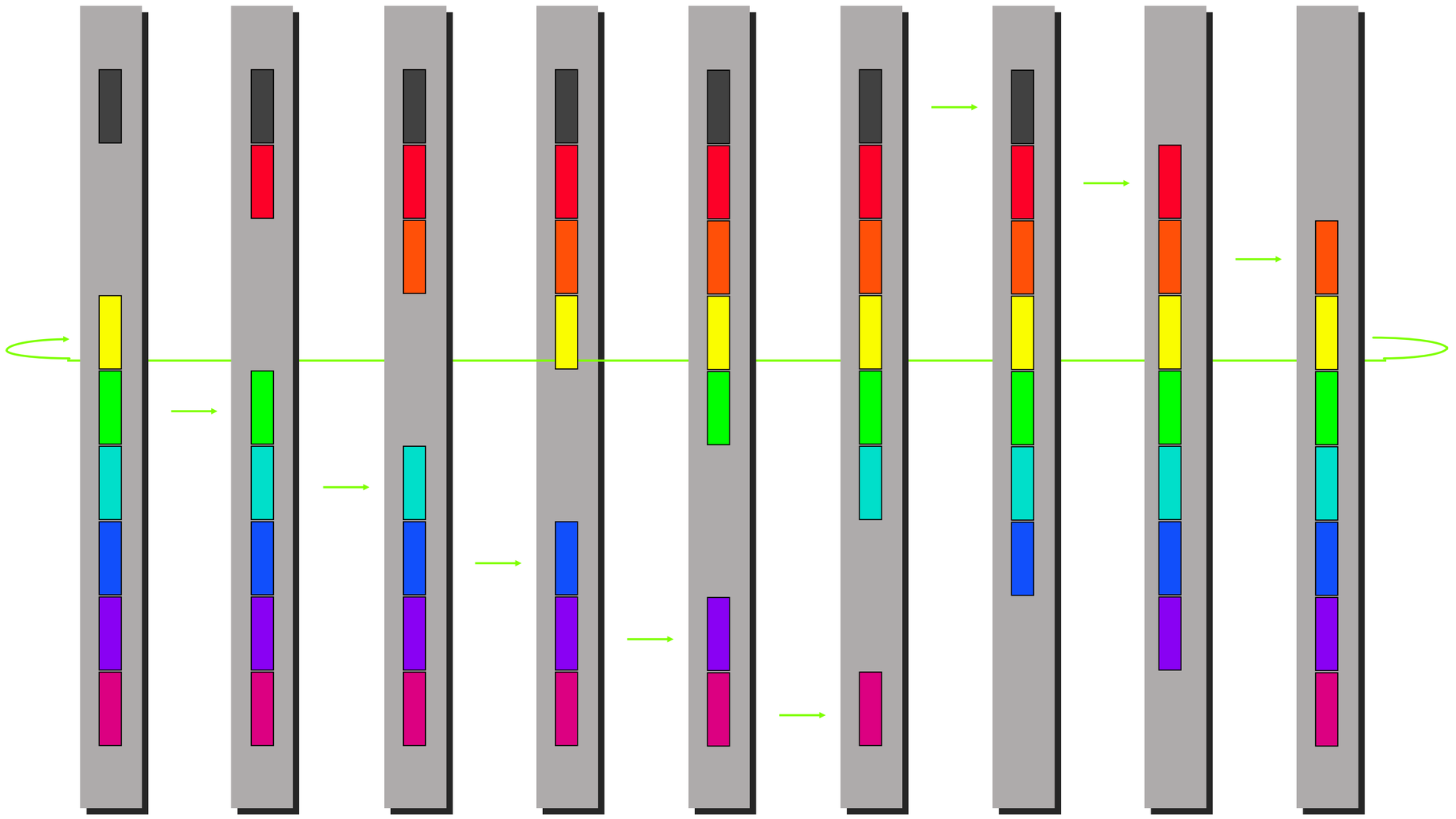


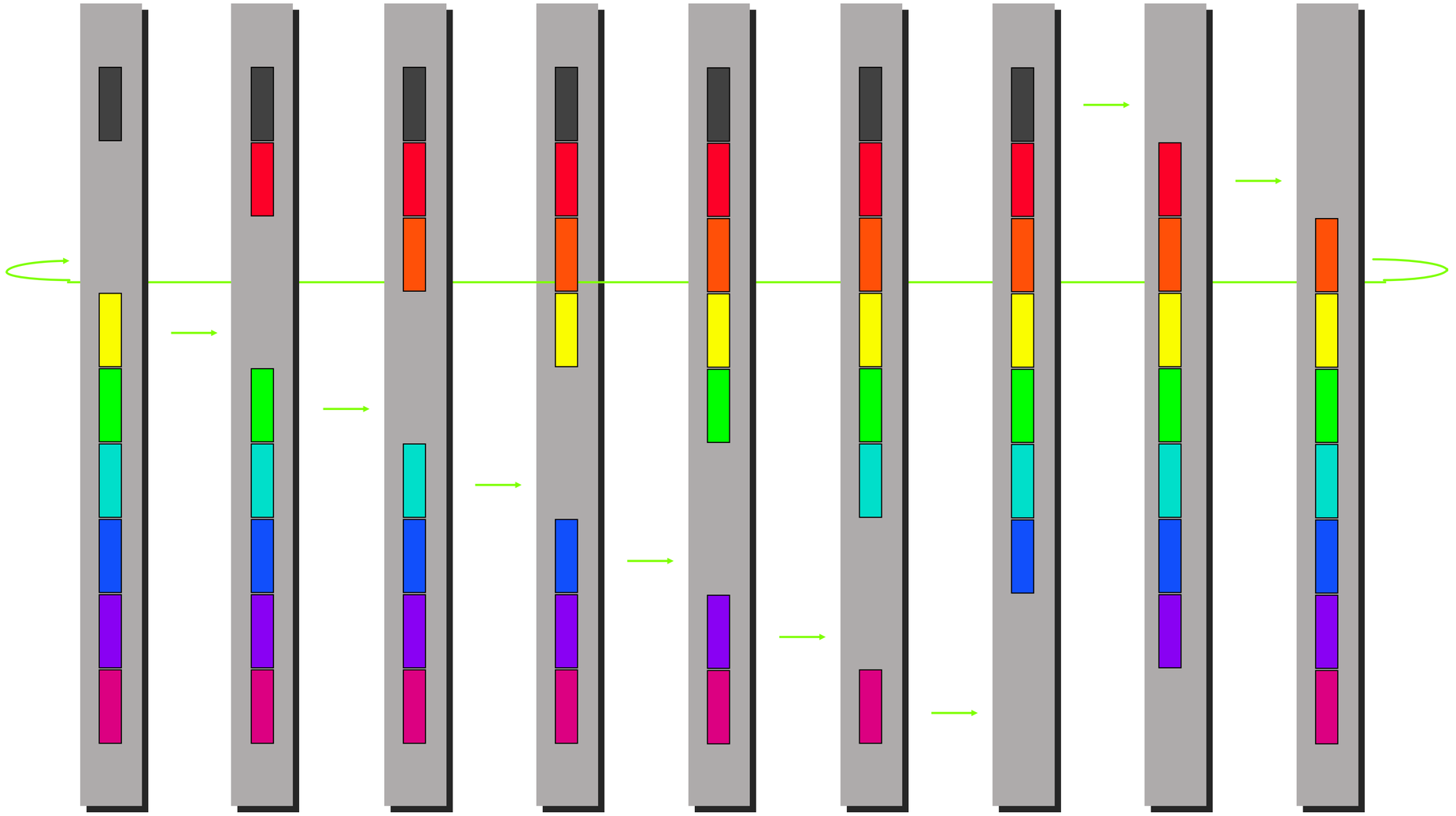


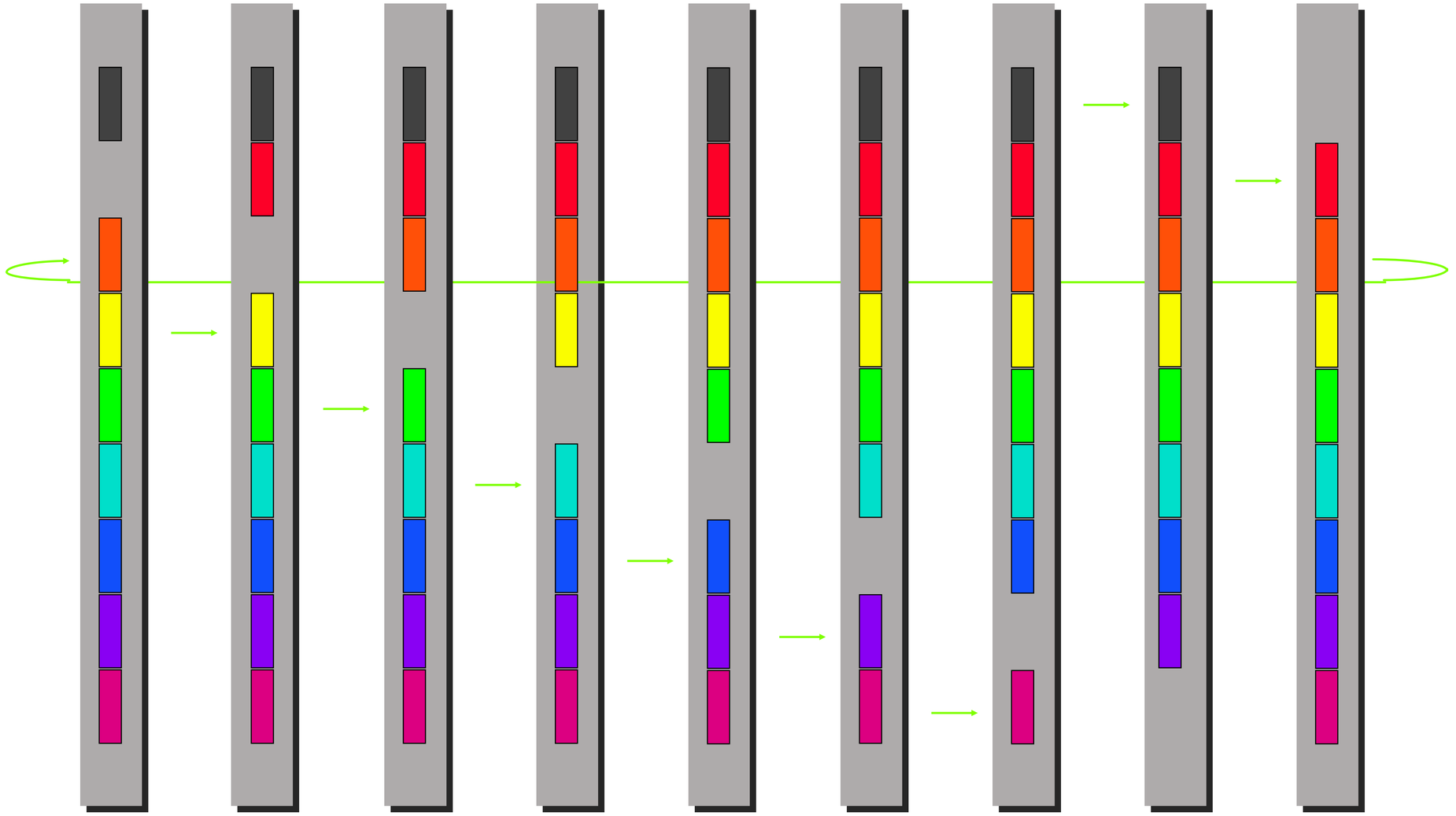


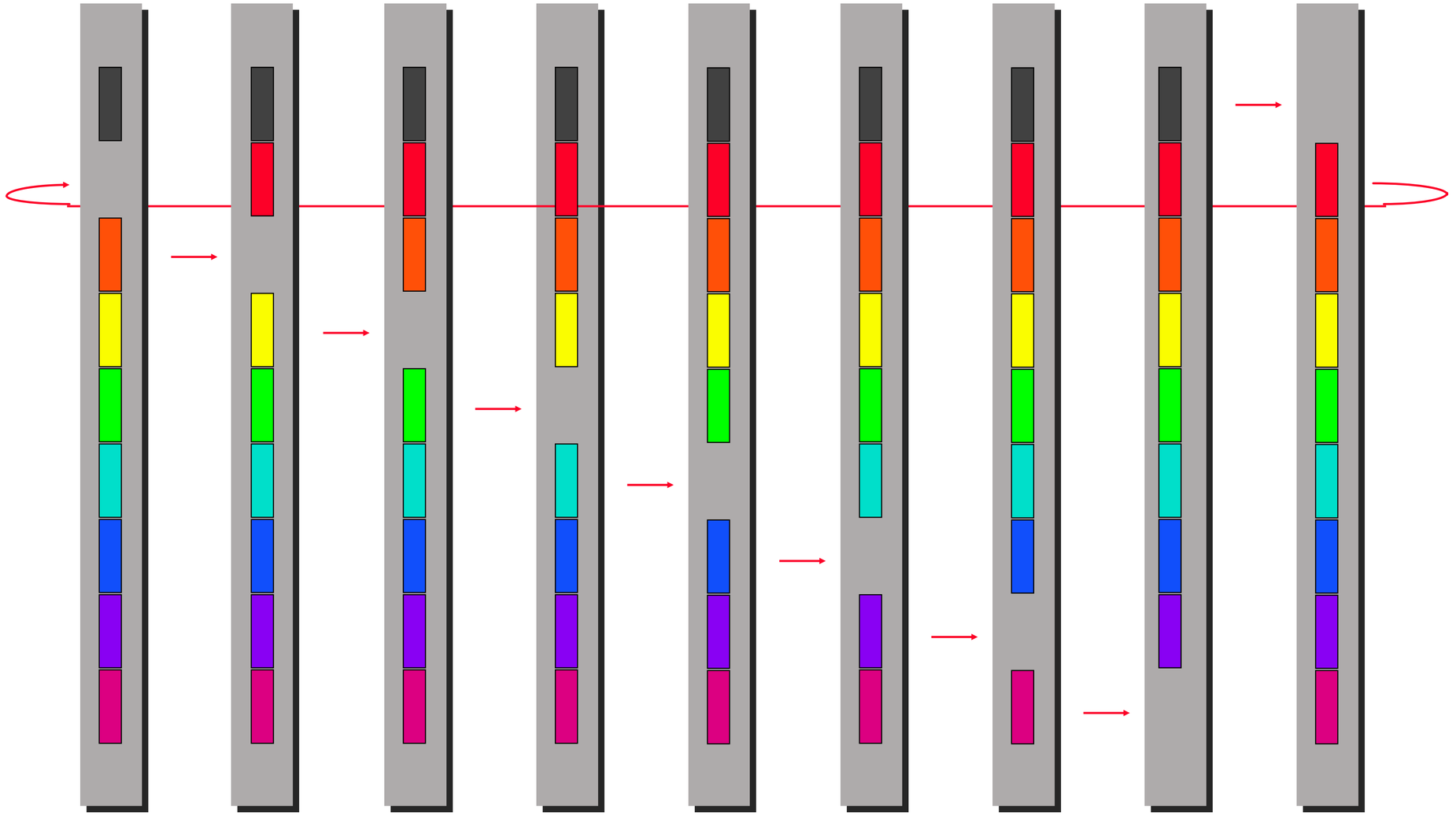


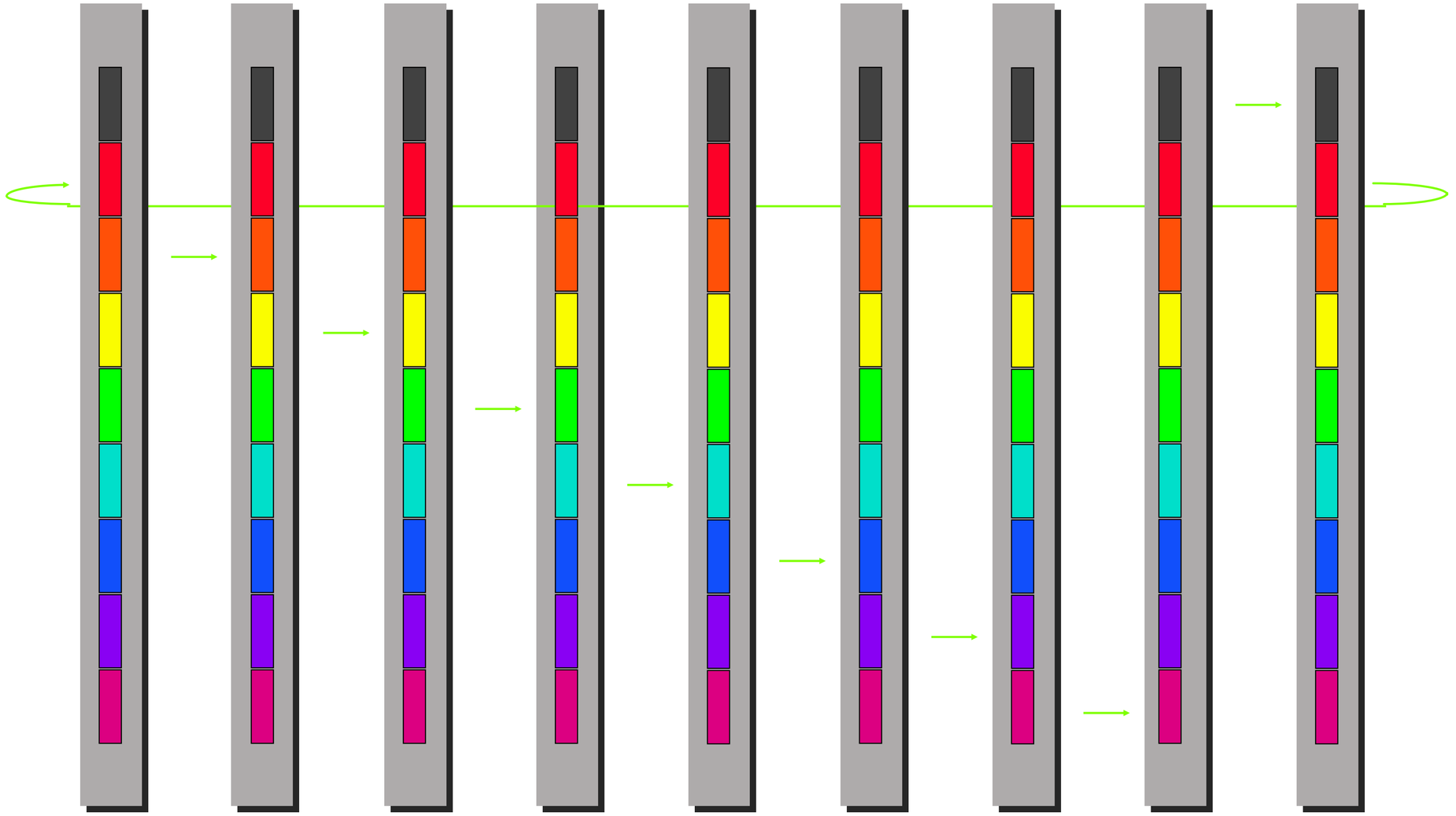


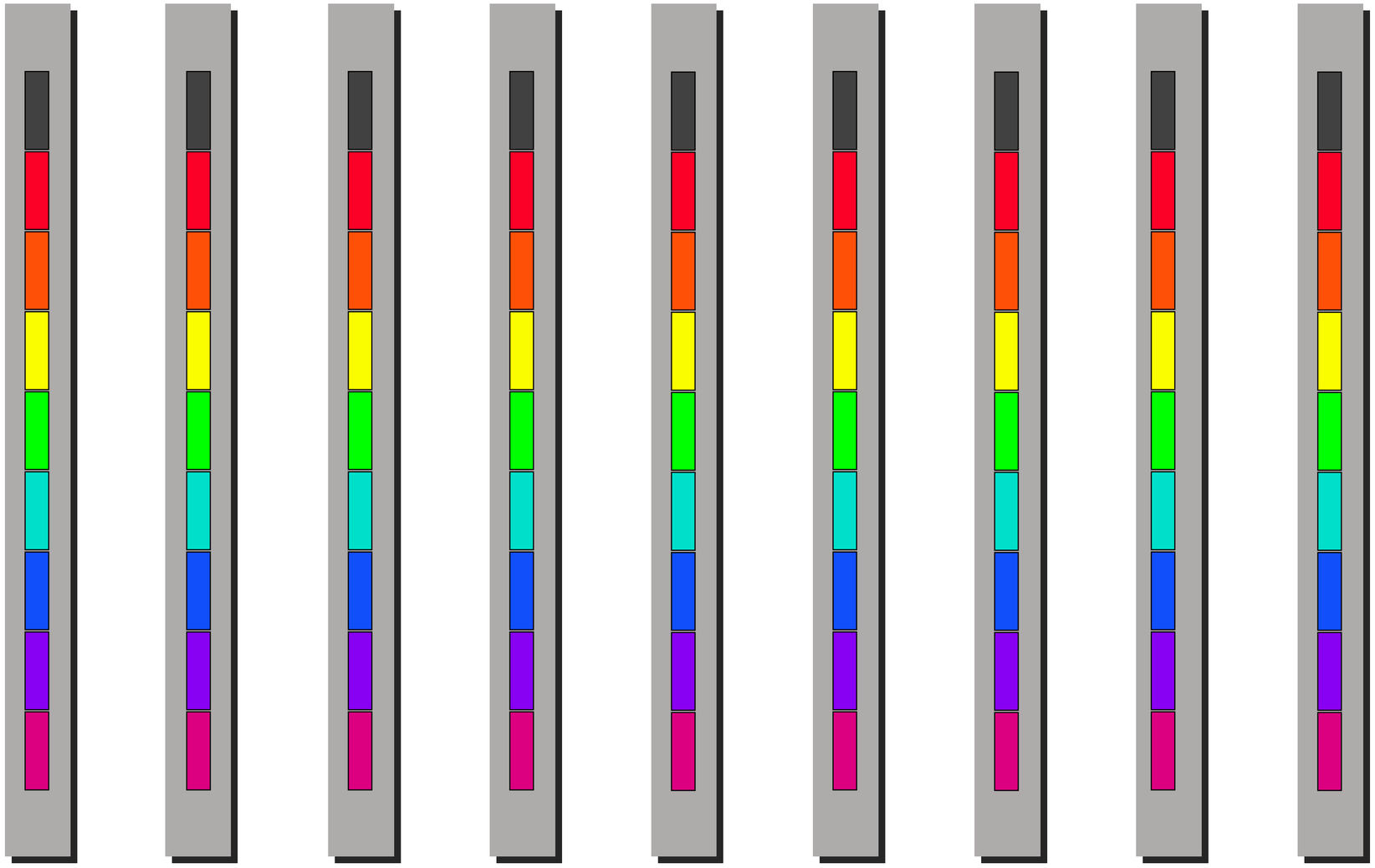




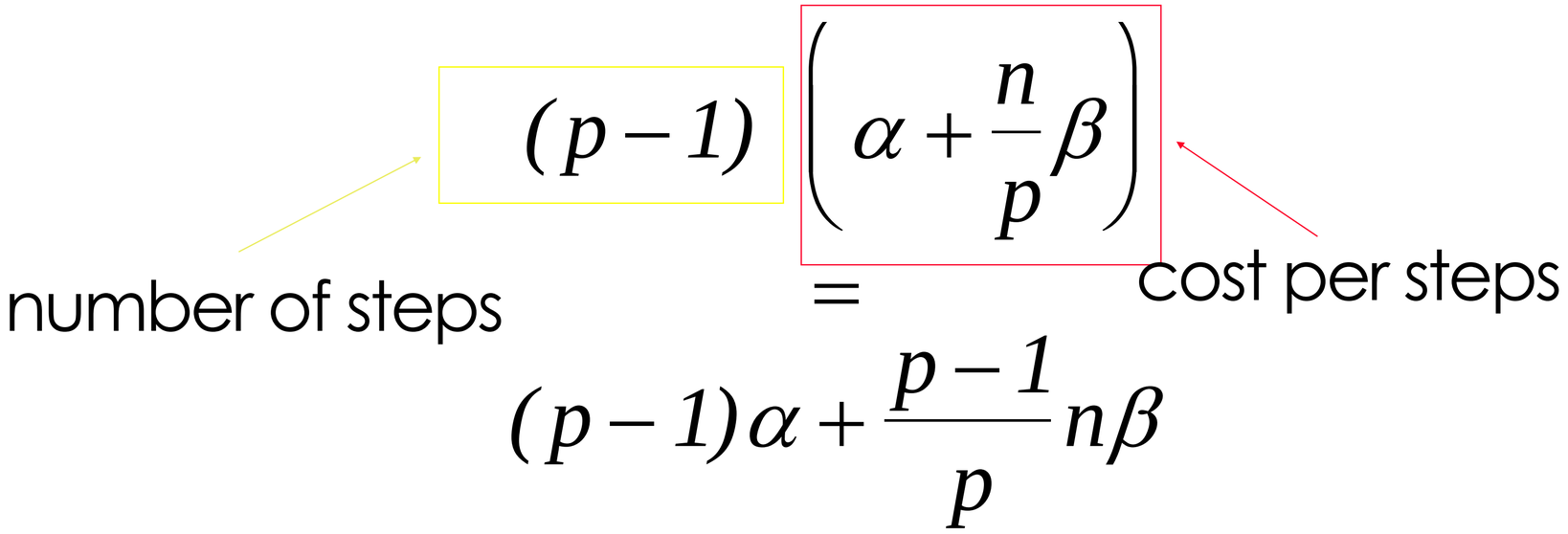




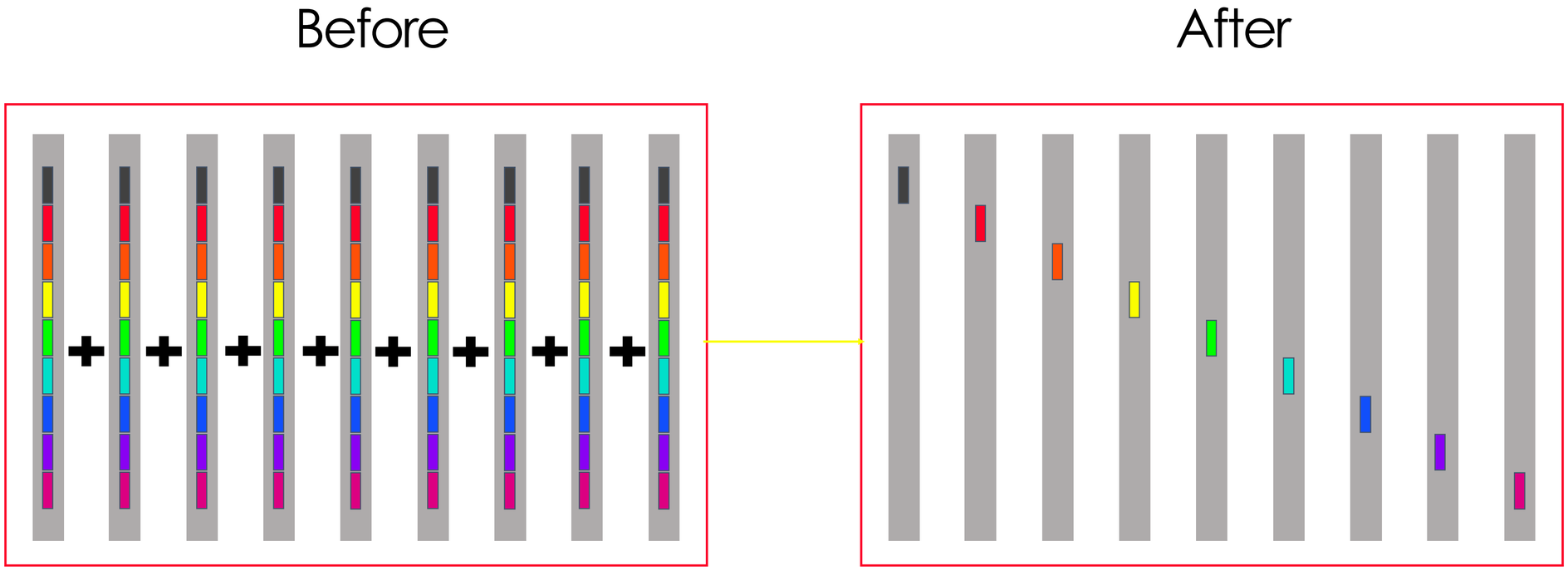


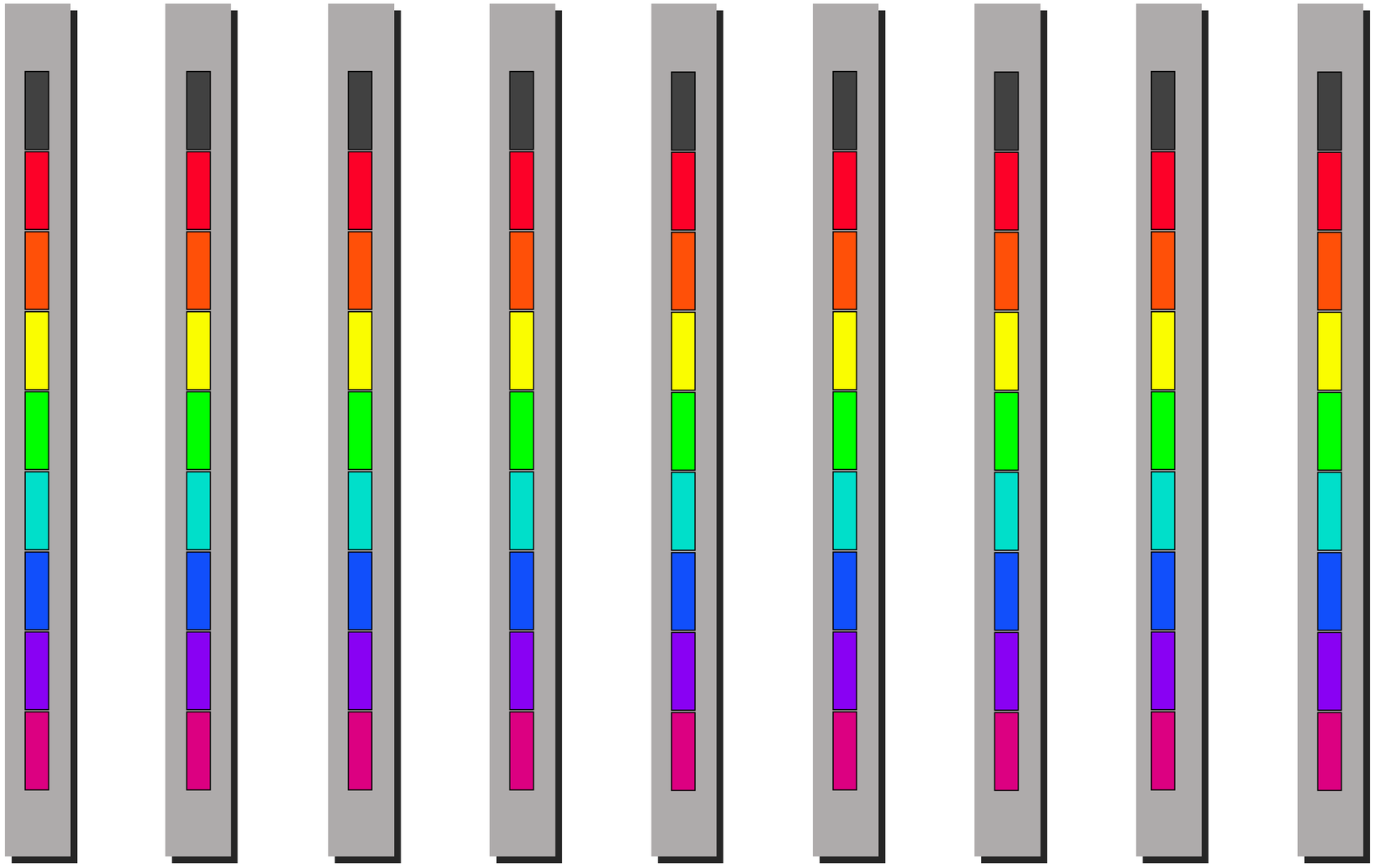


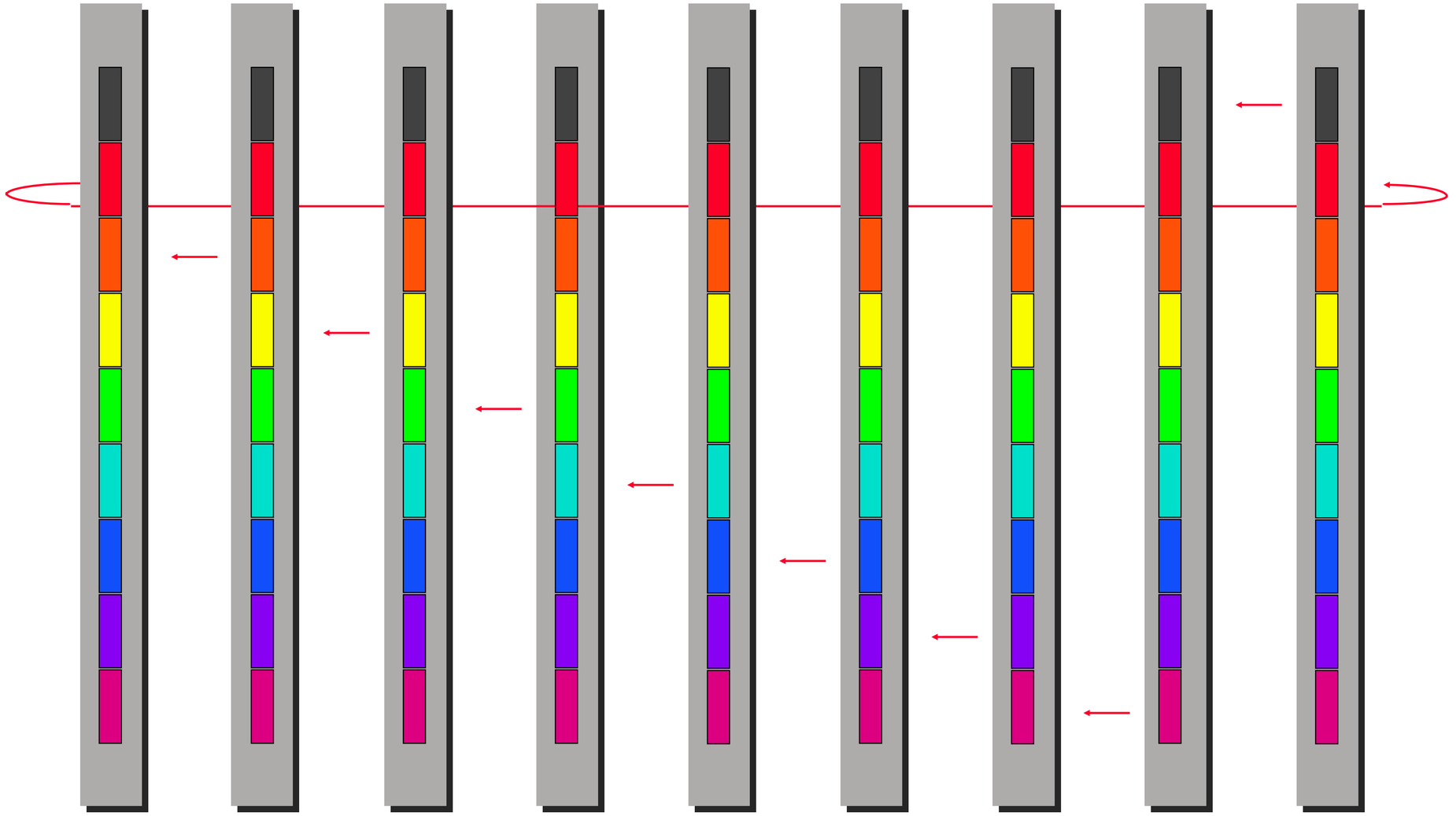
Cost of bucket Allgather

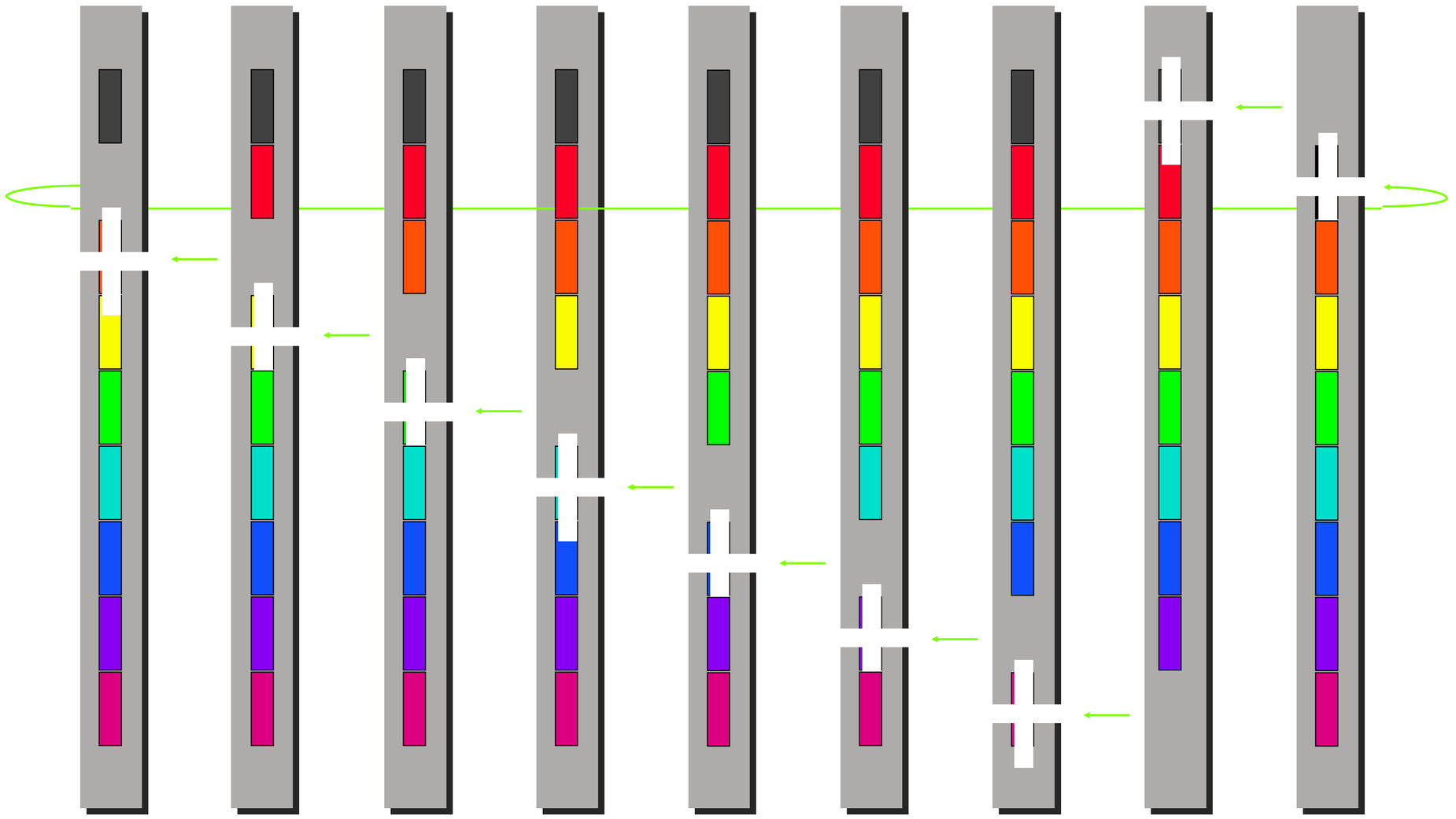


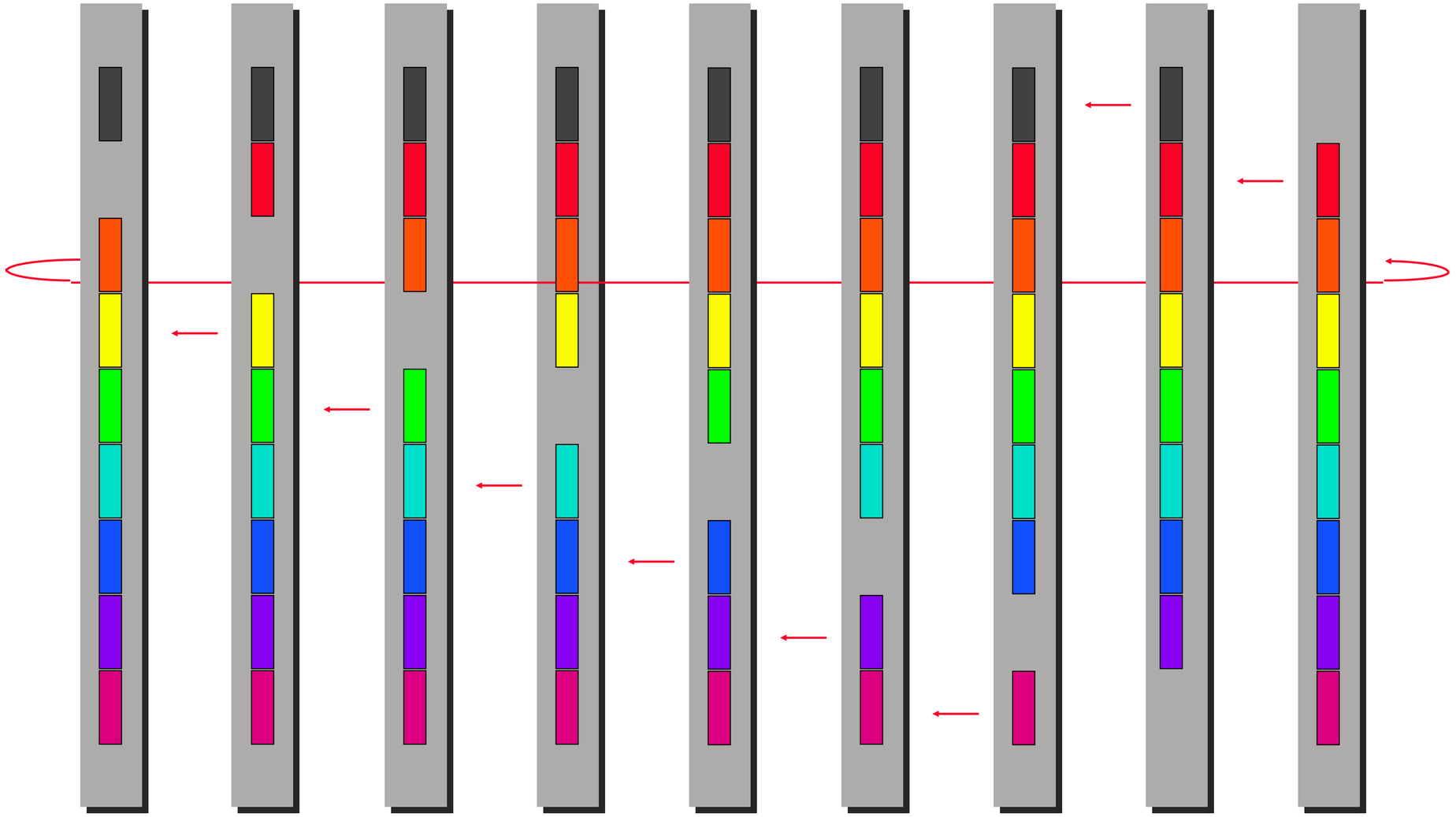
Reduce-scatter

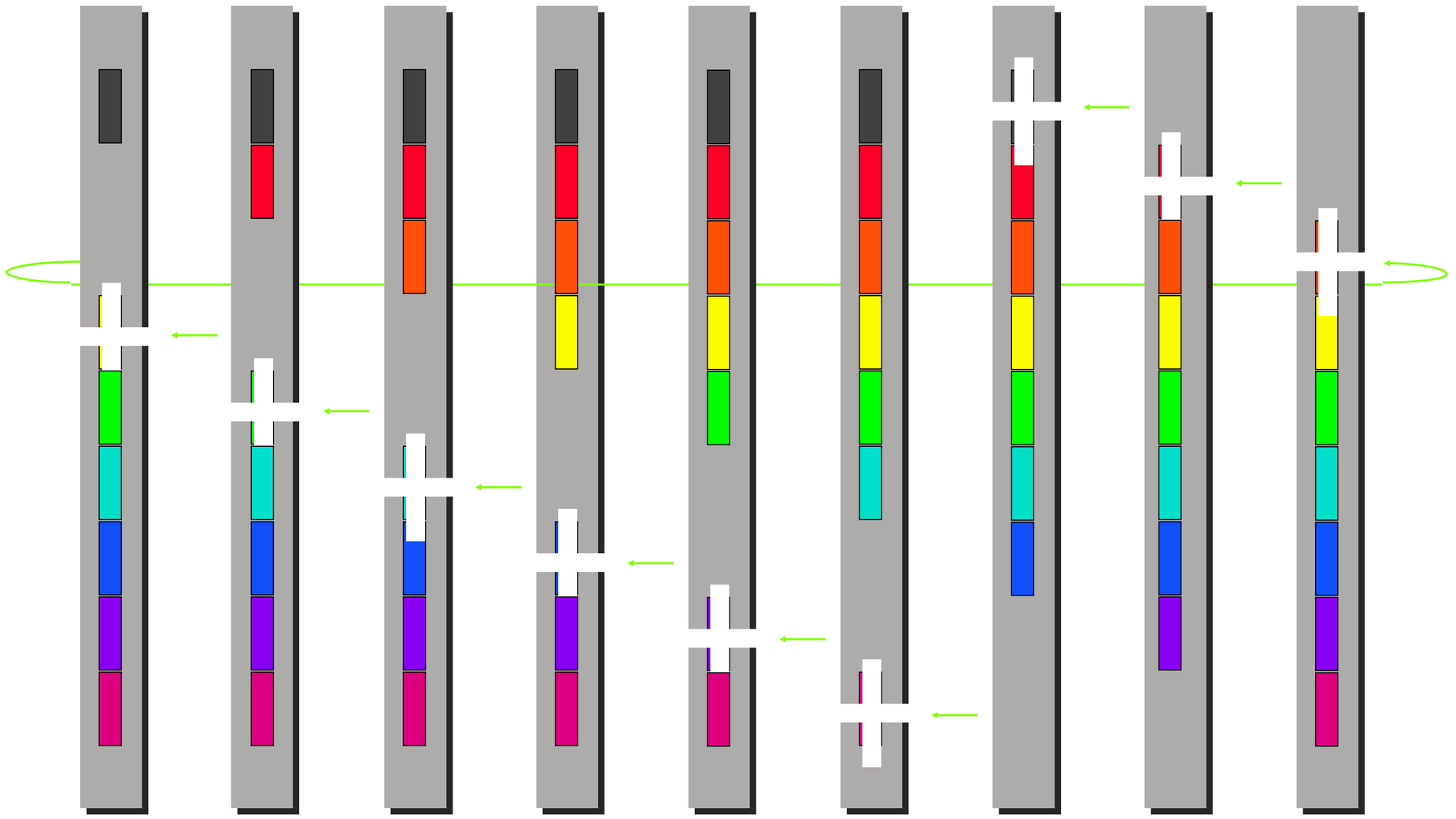


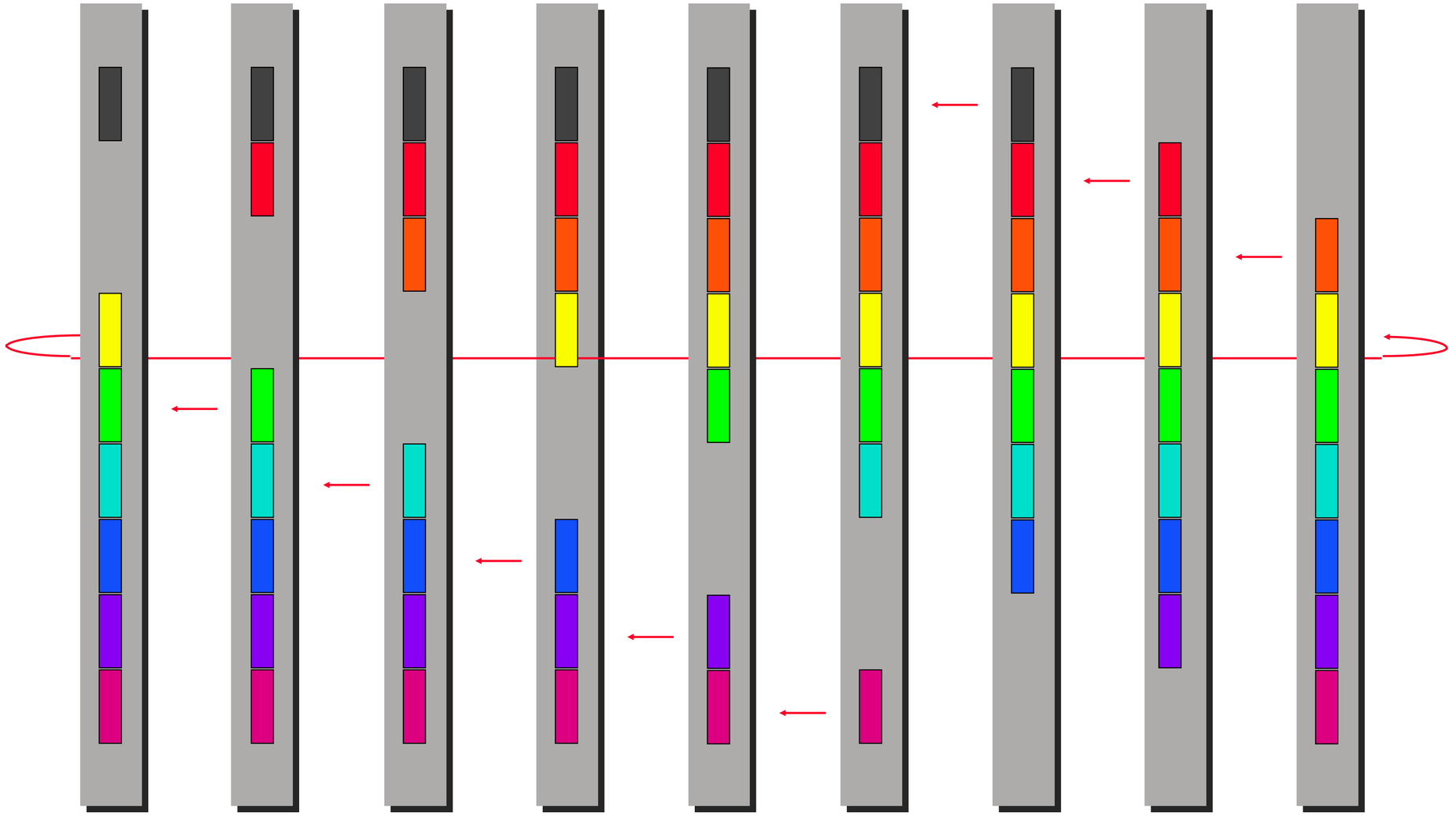


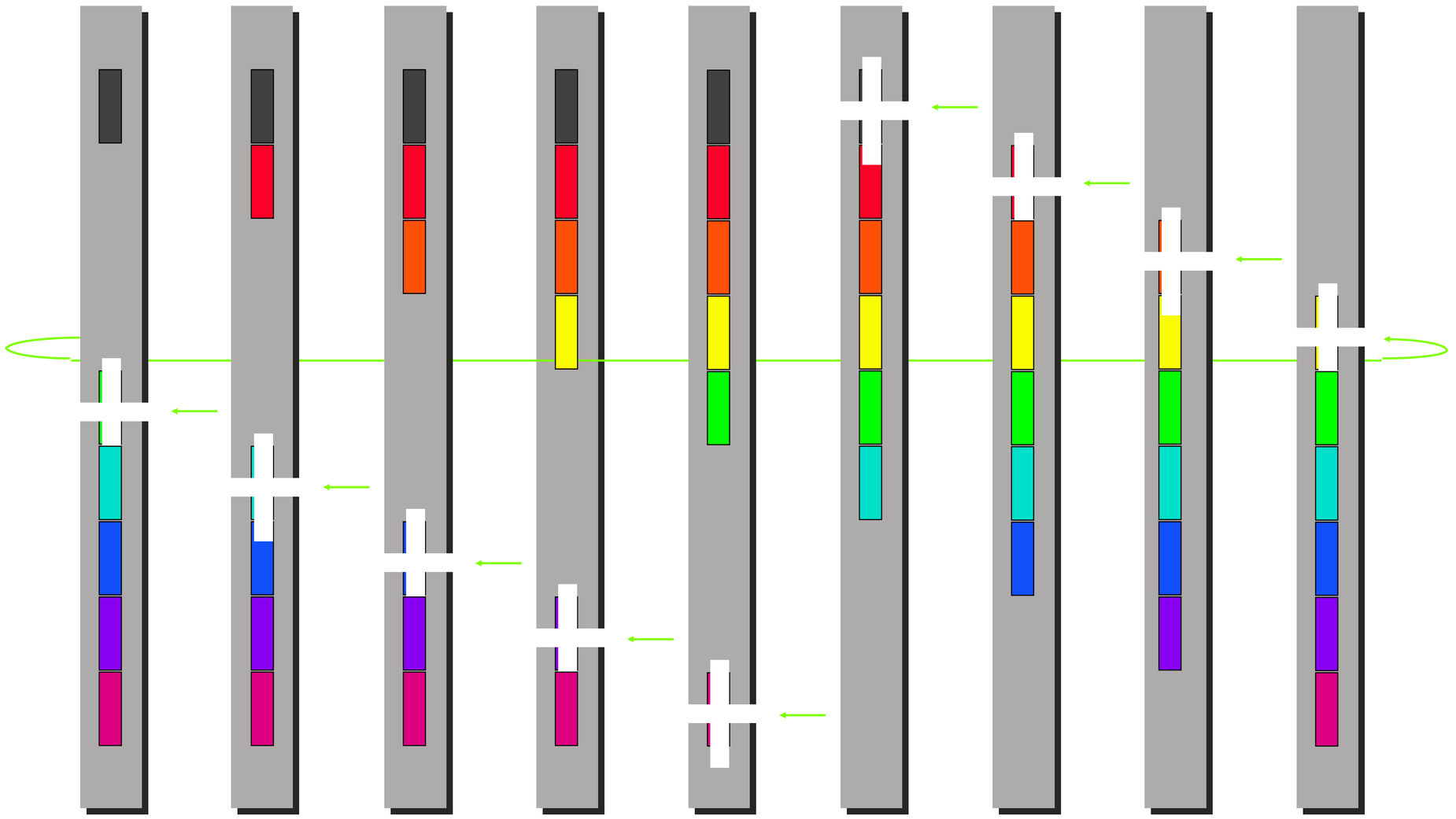


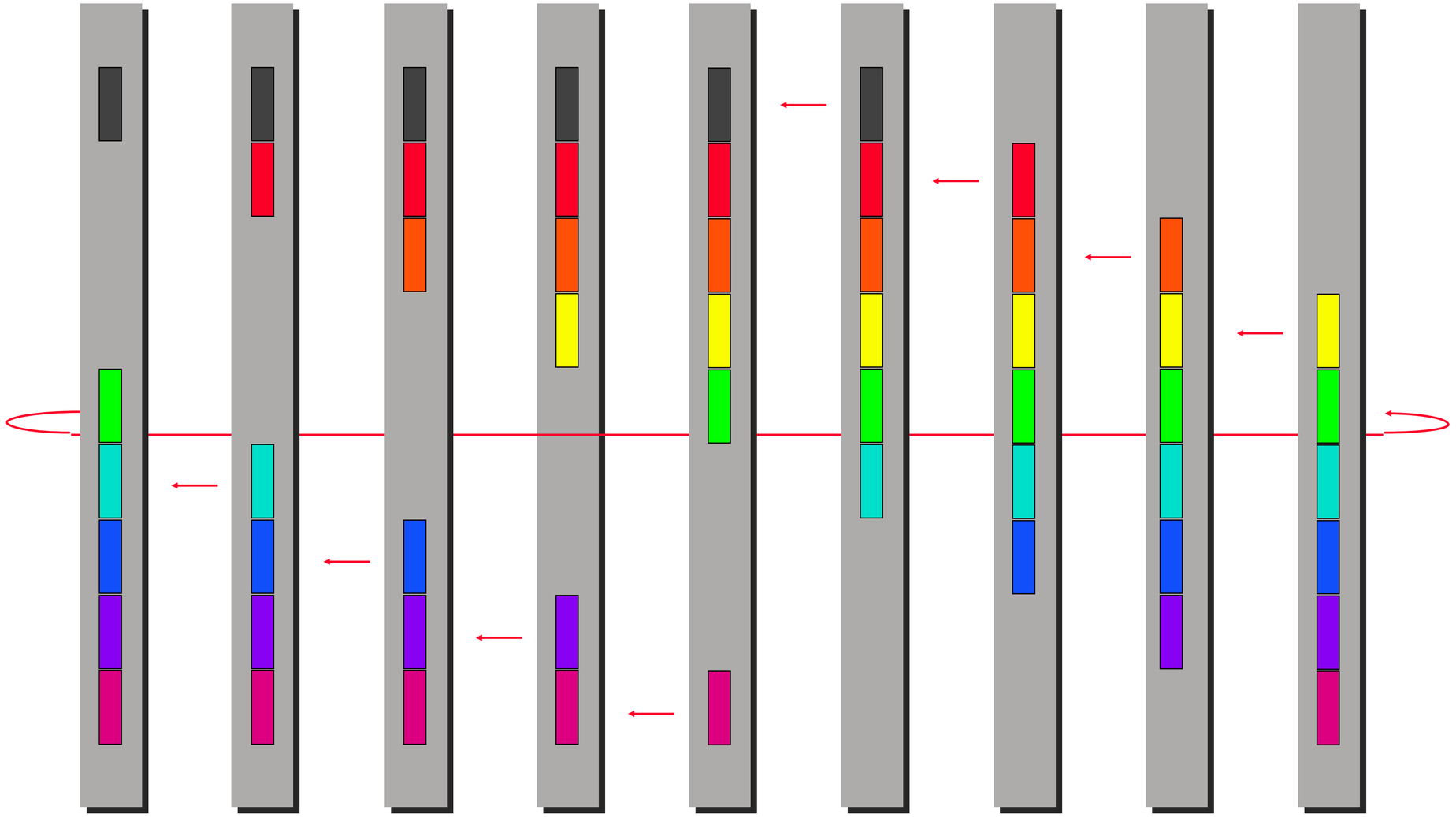


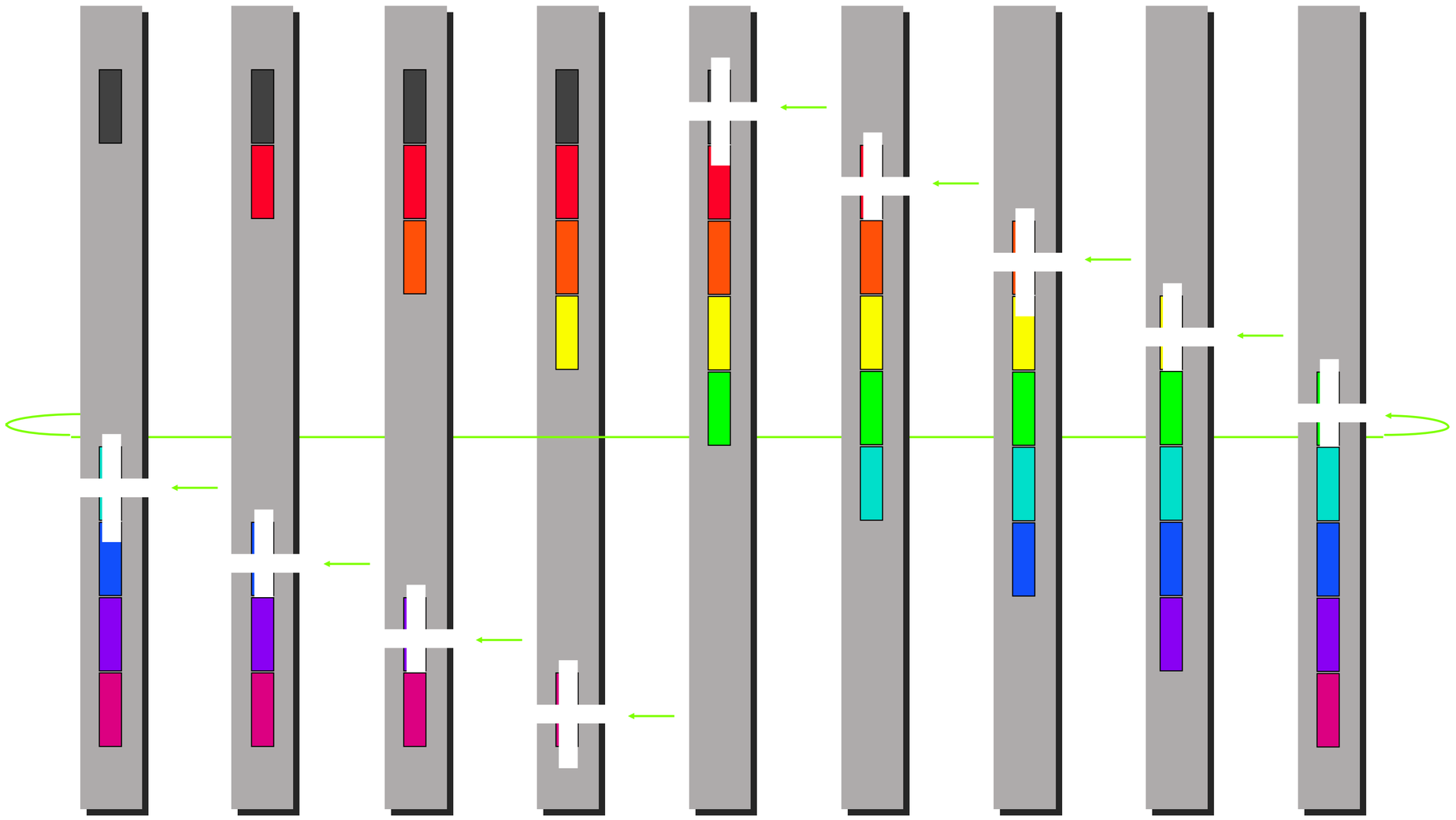


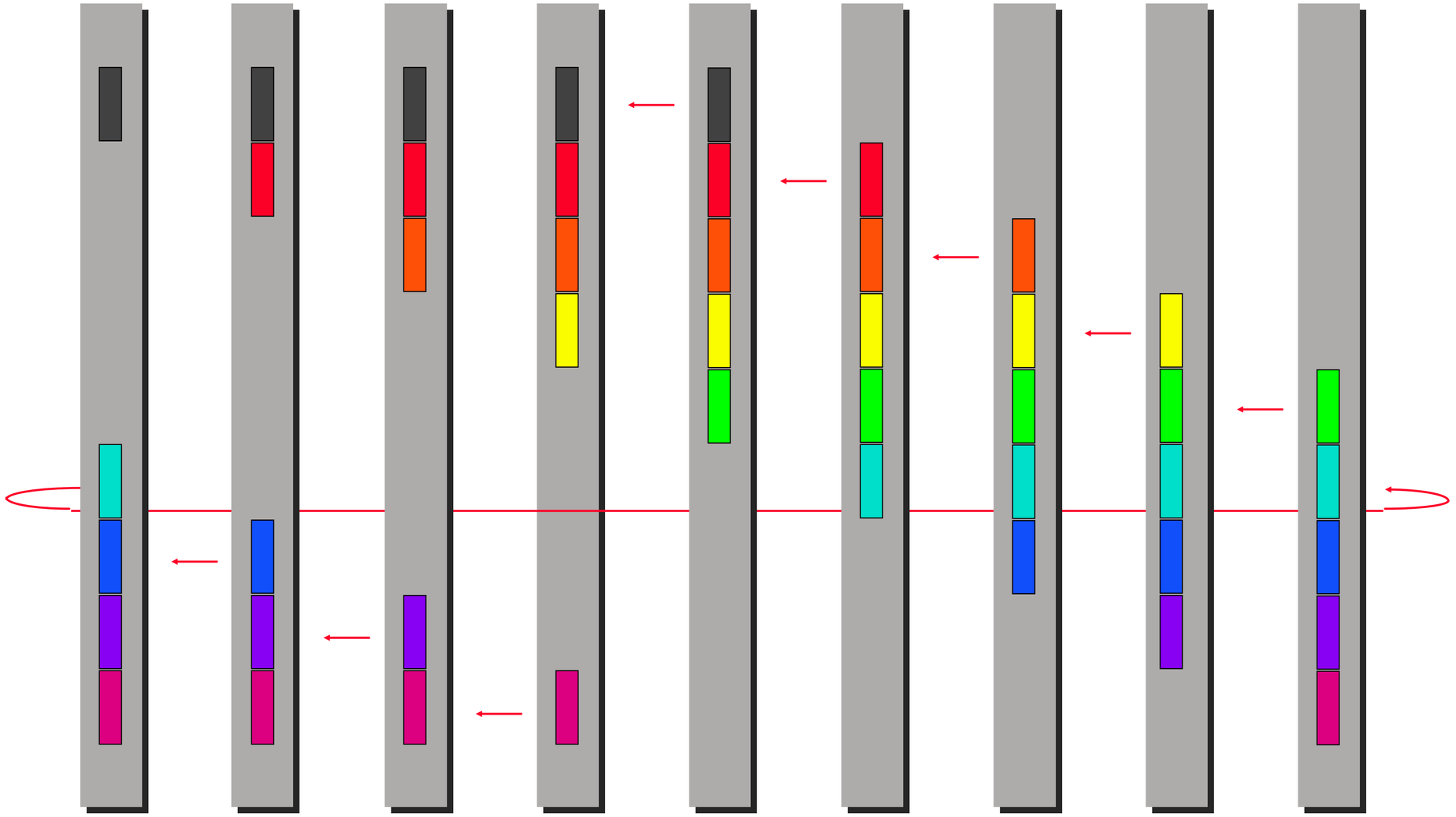


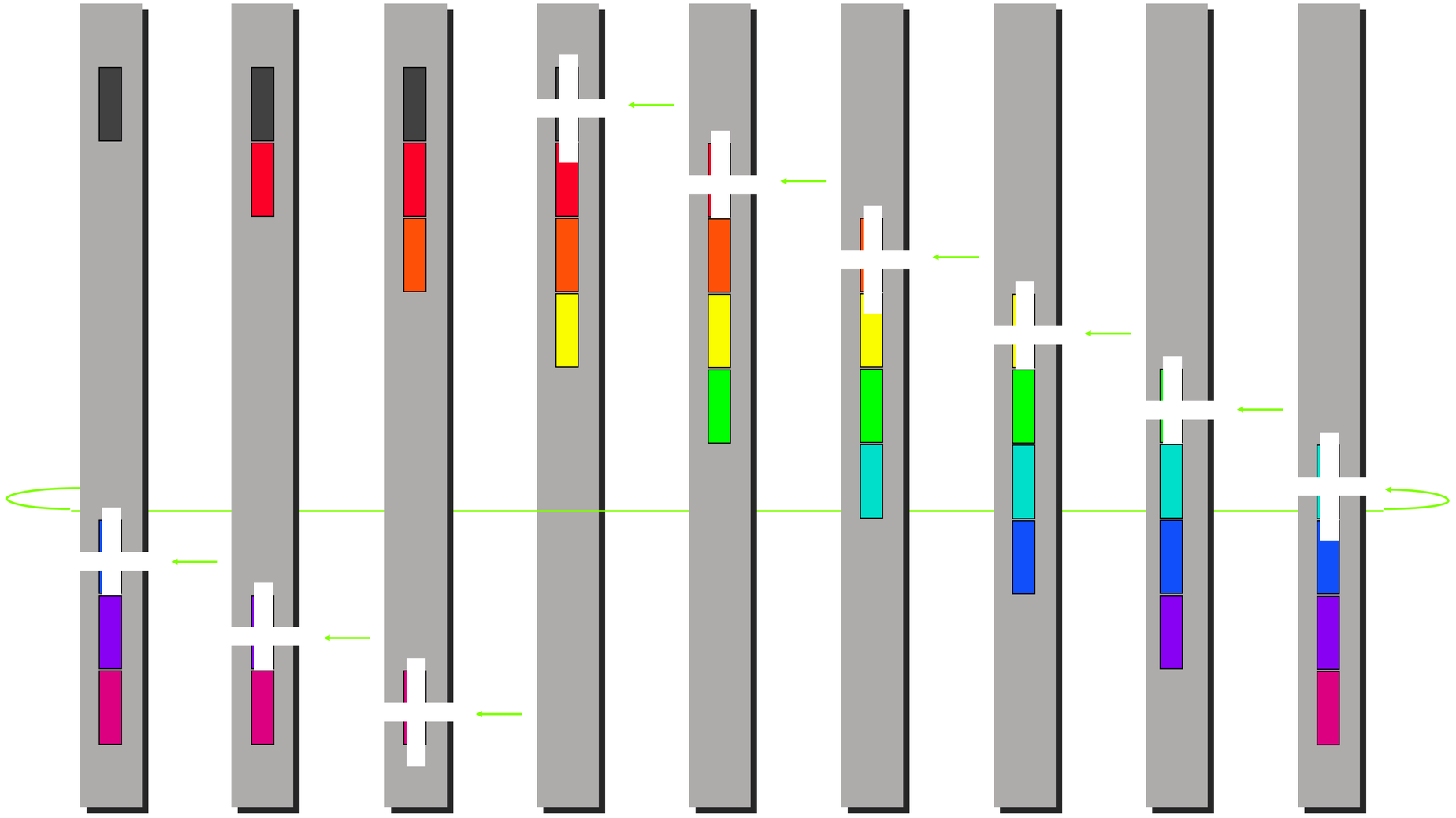


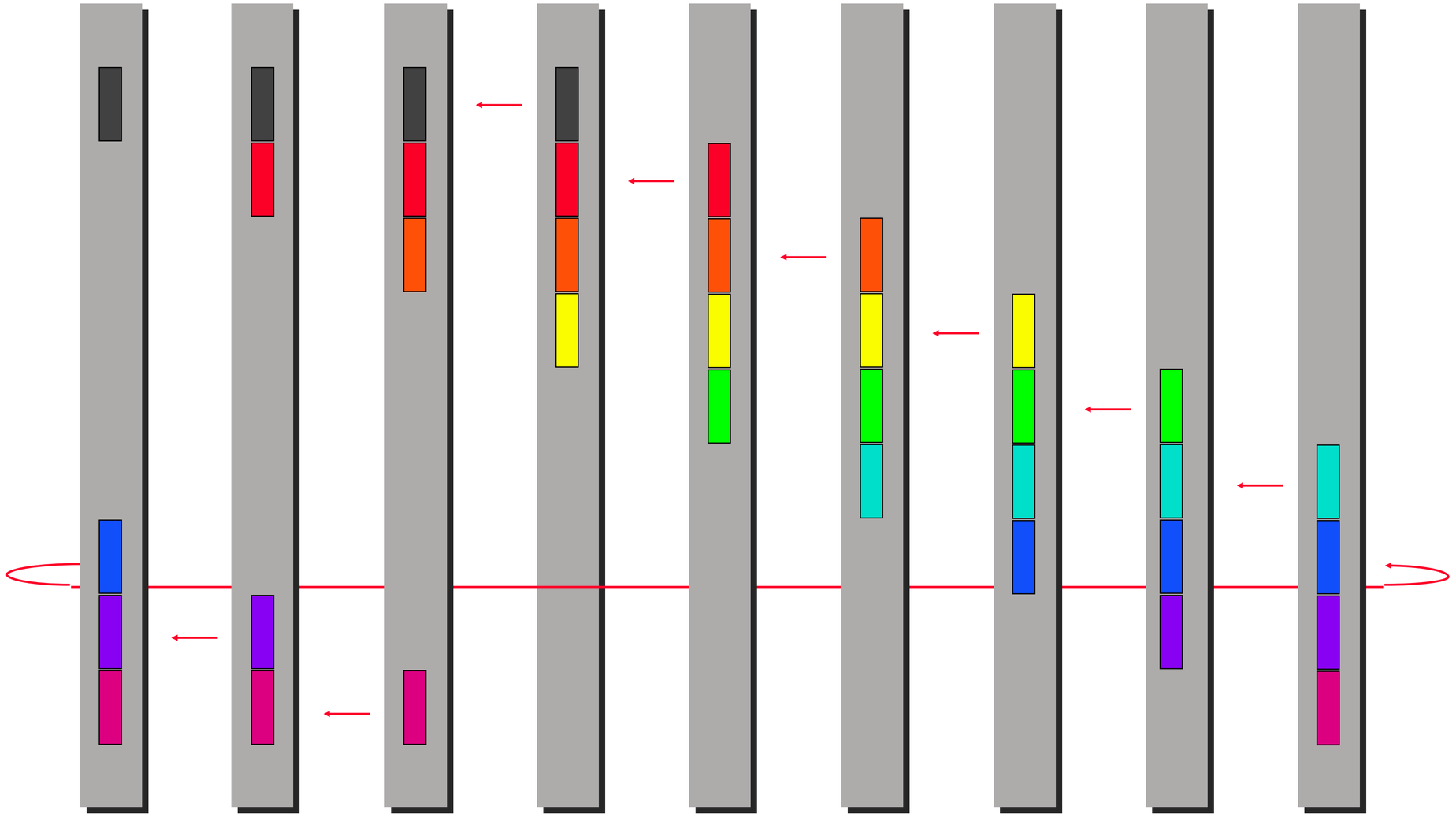


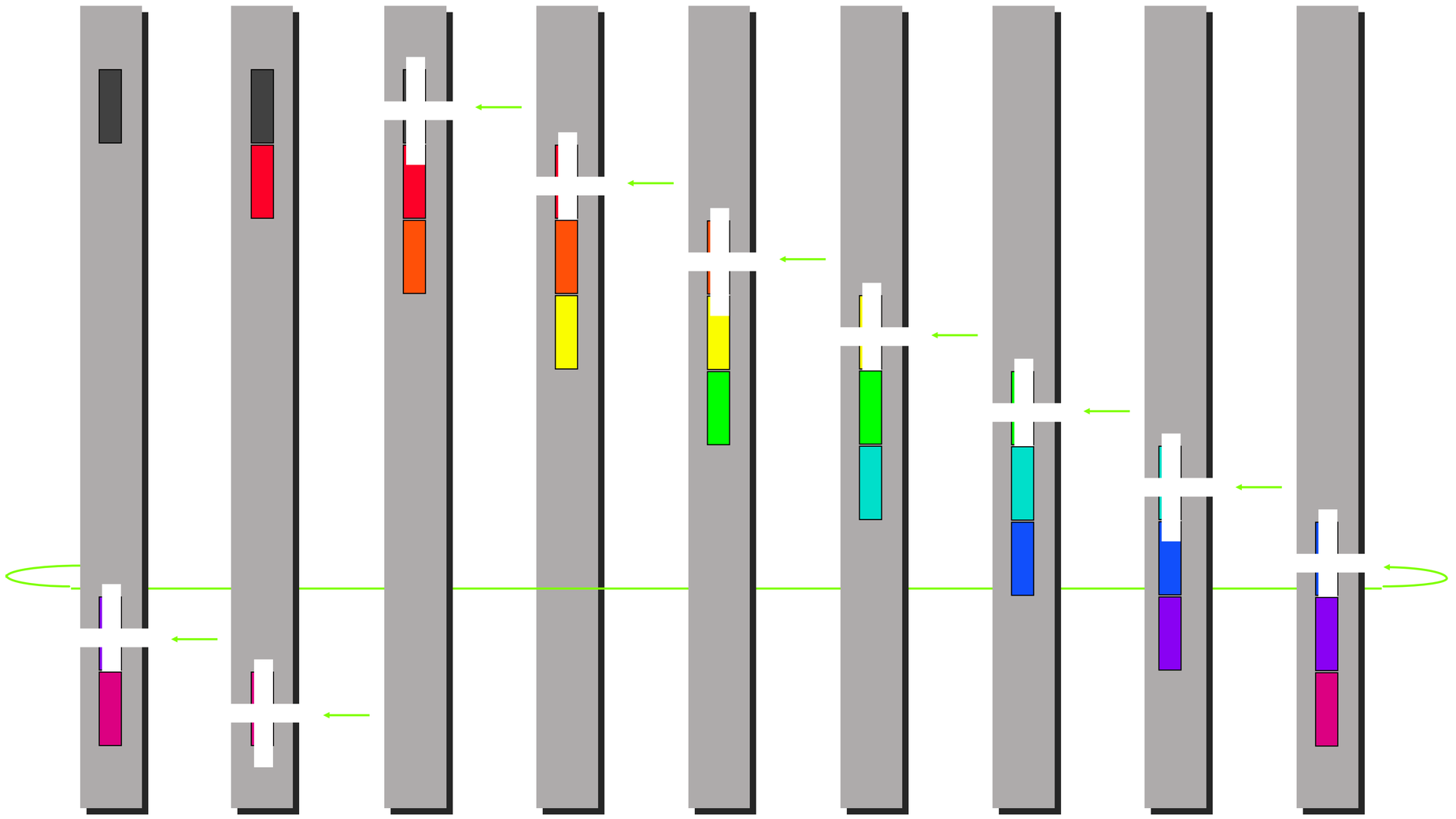


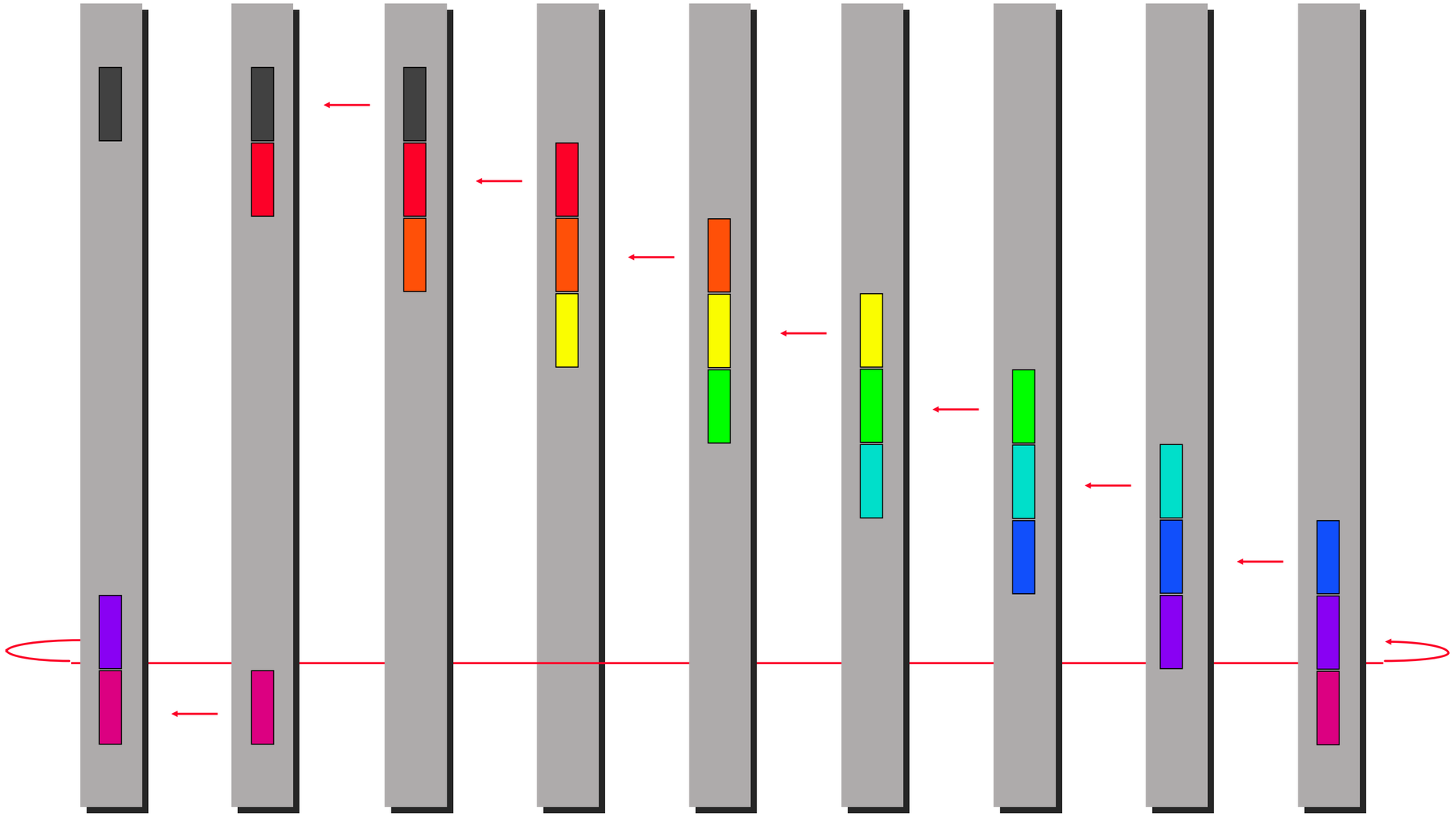


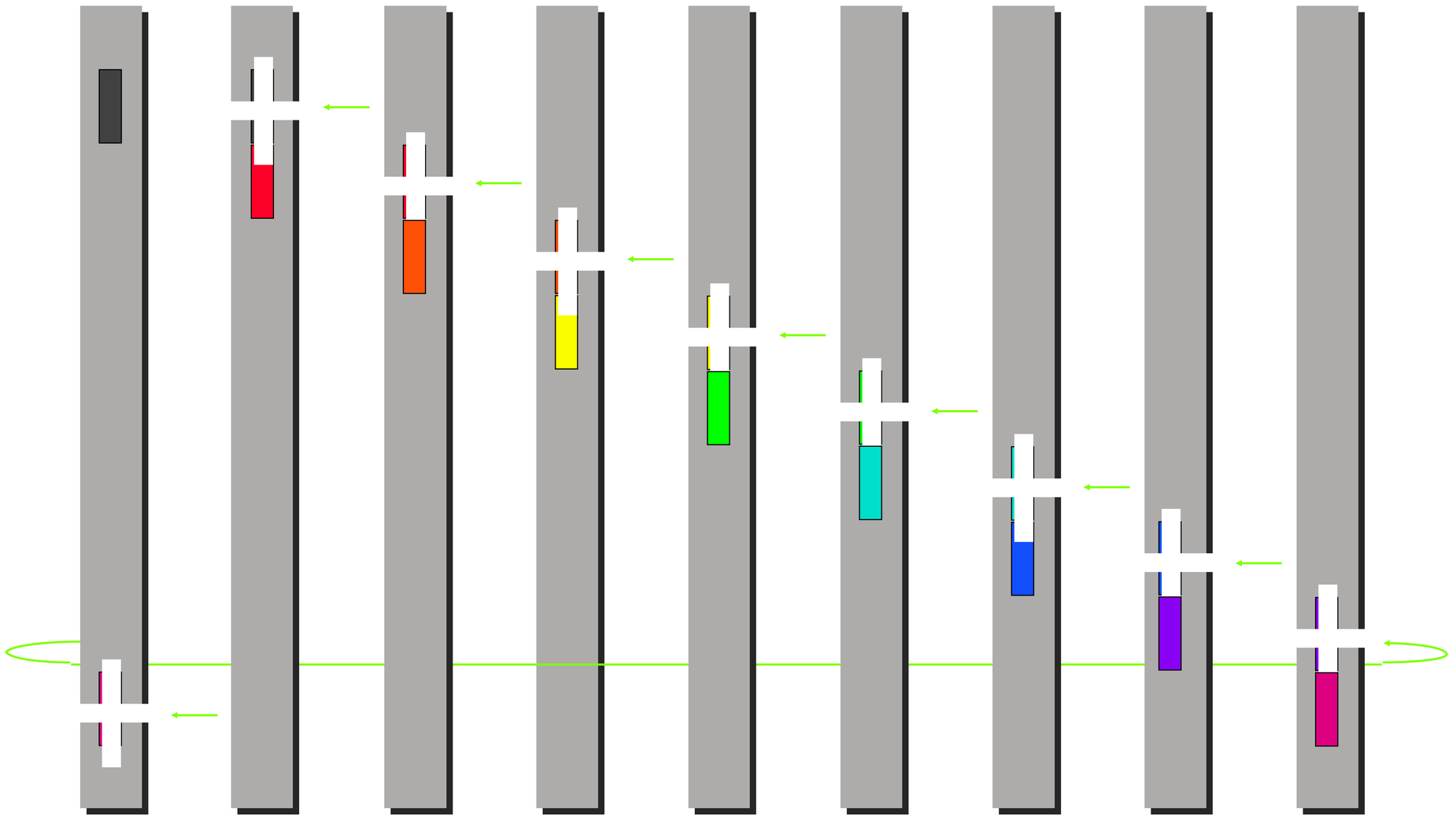


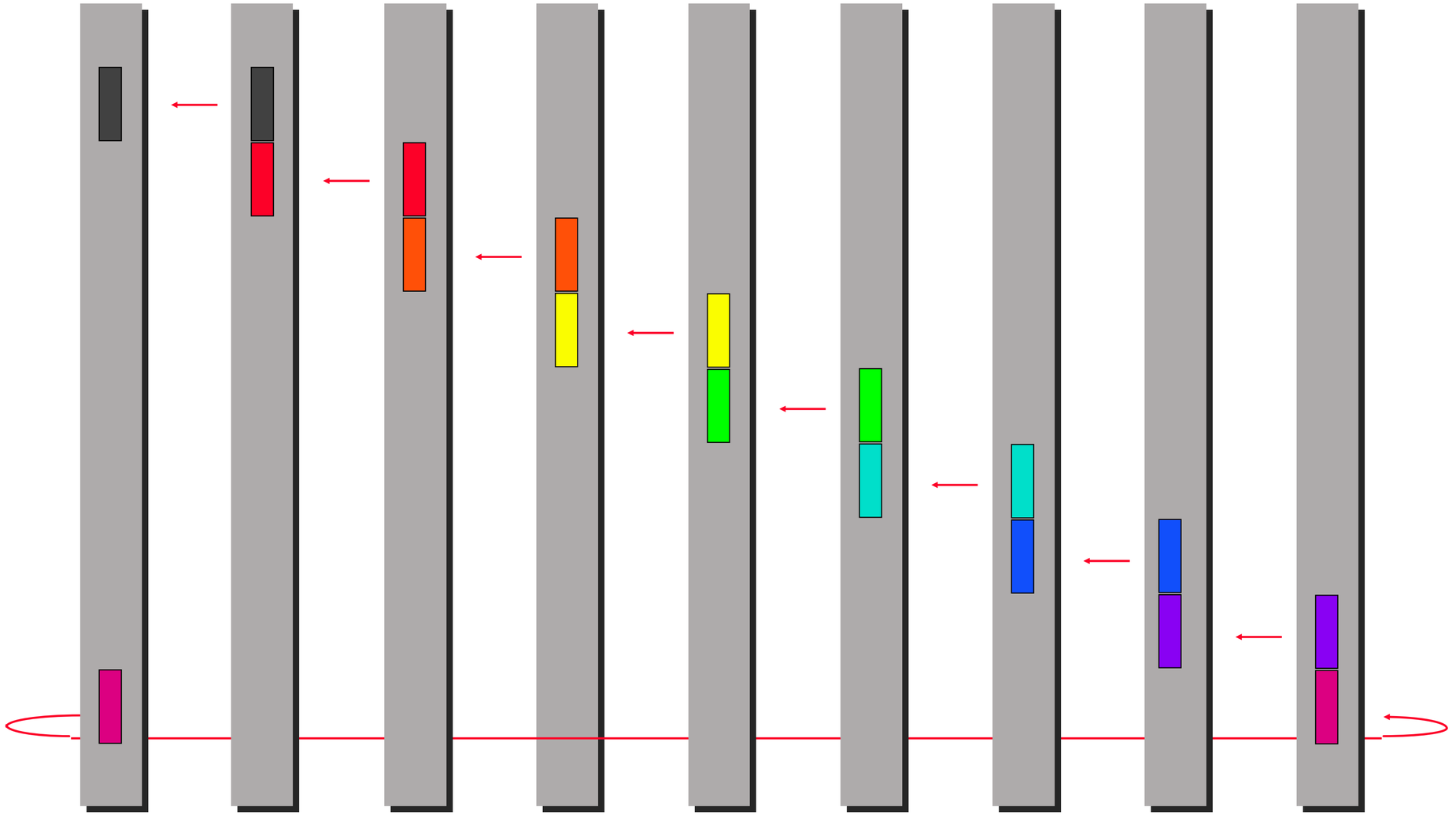


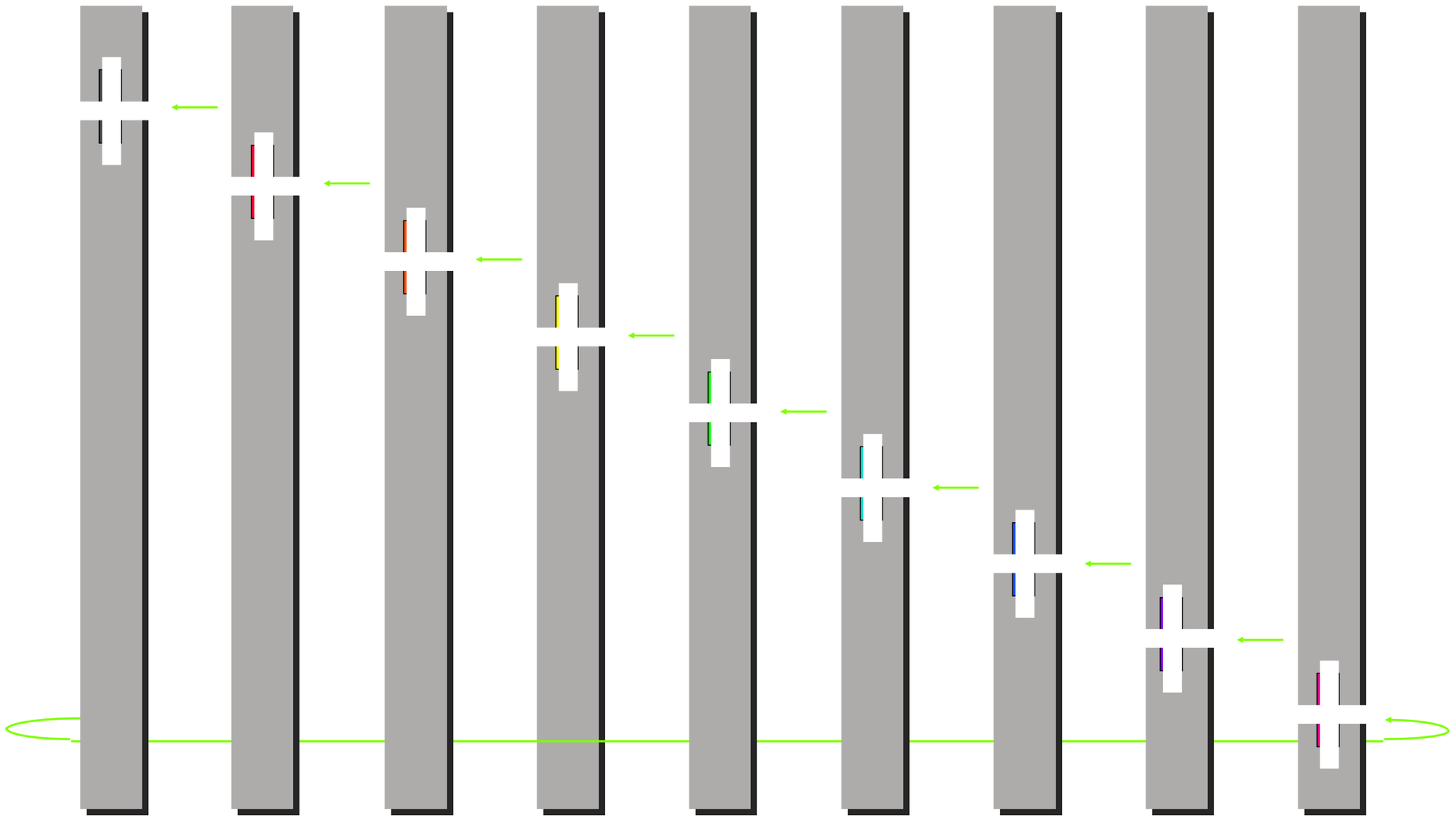


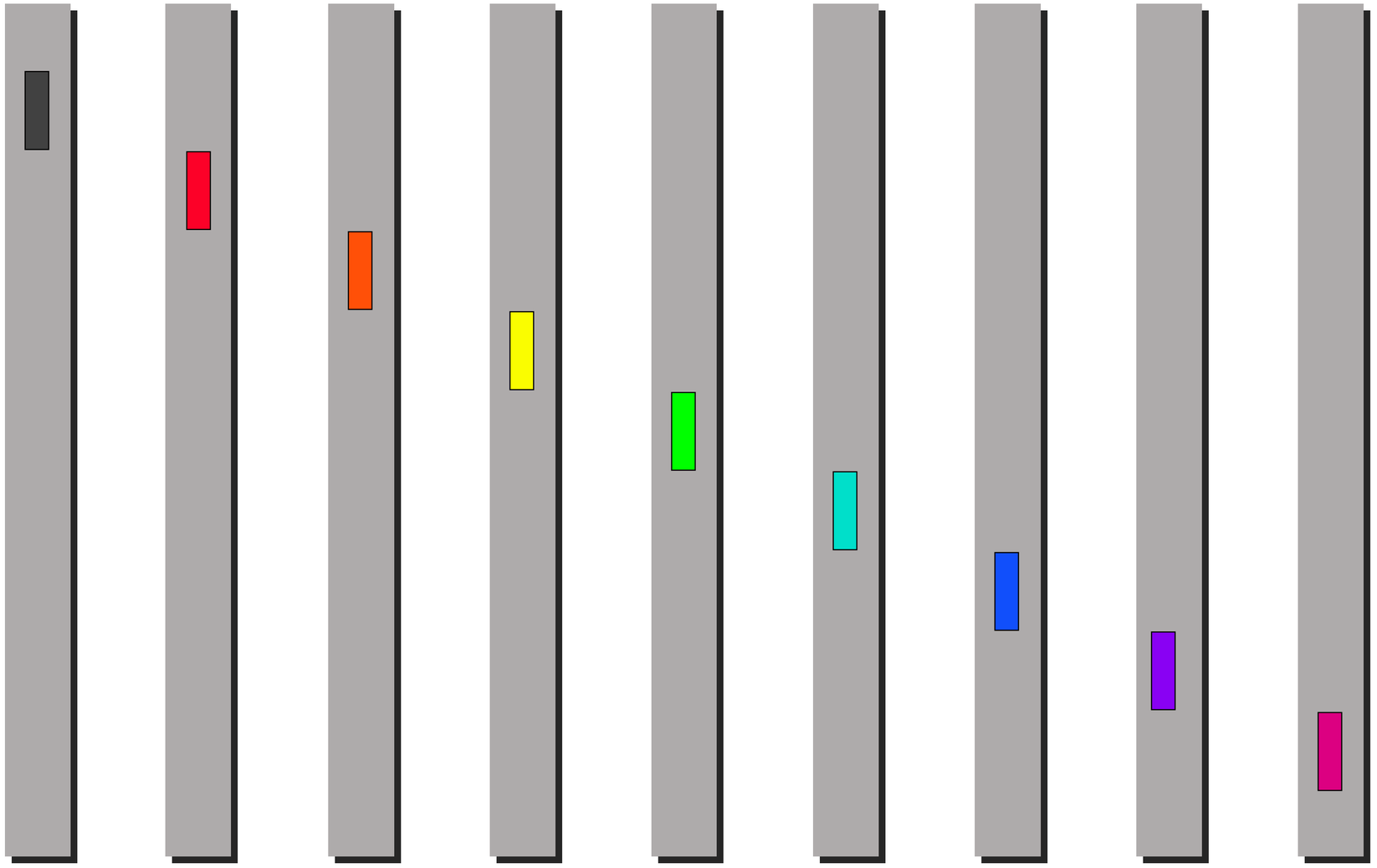




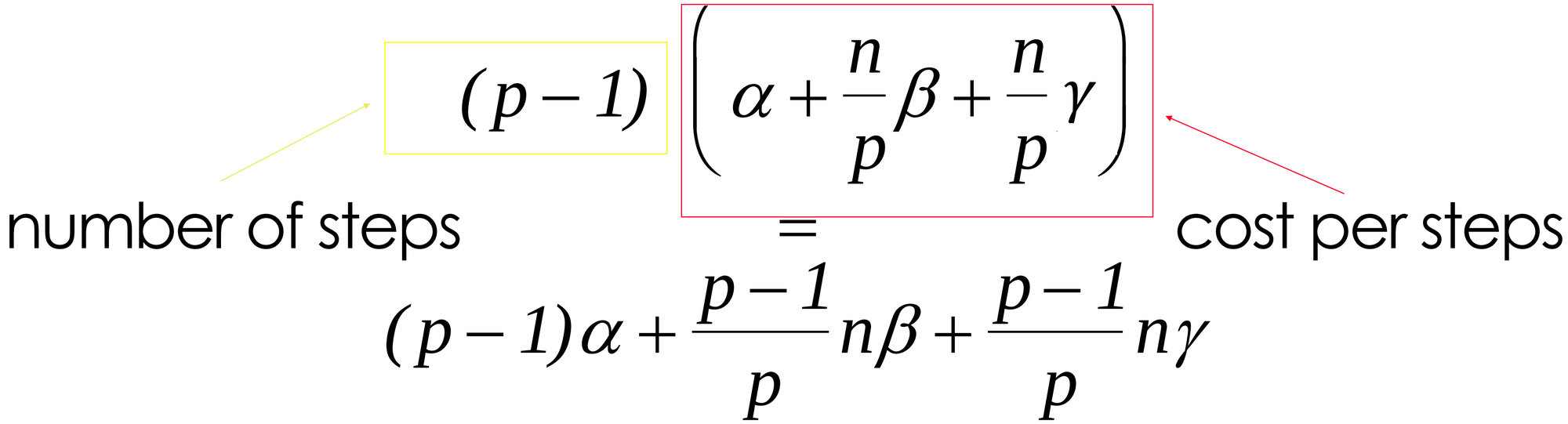






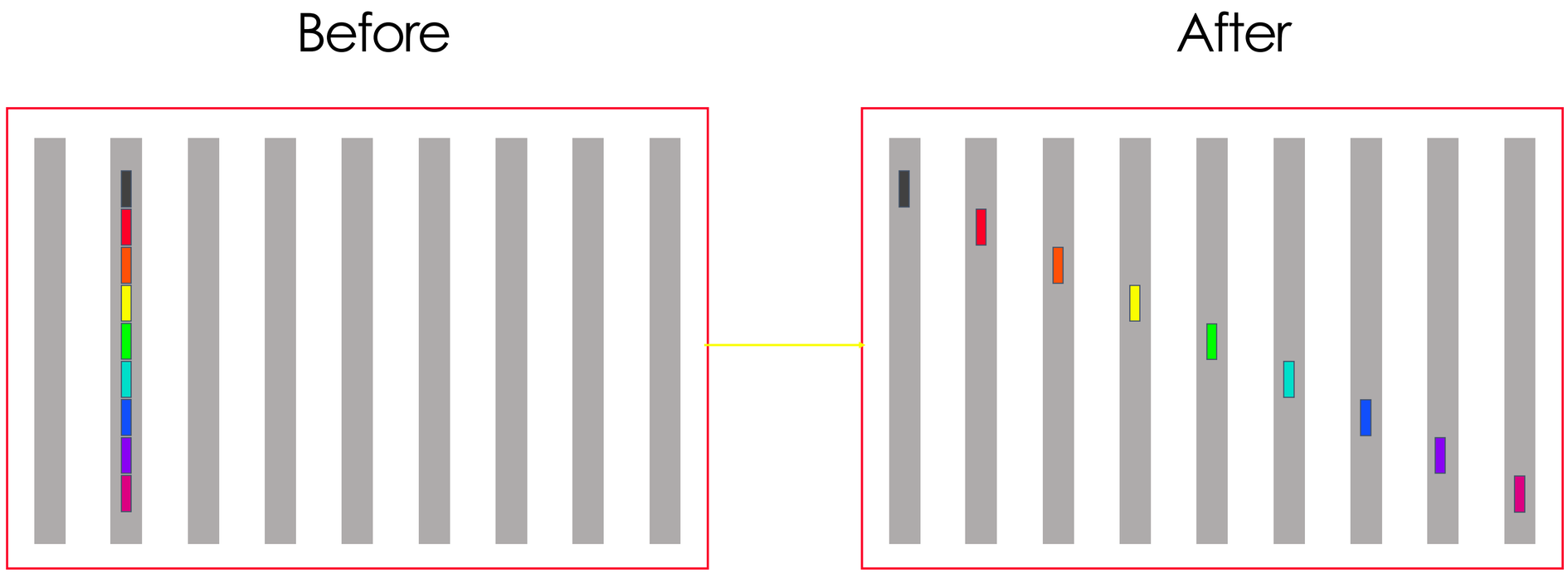


Cost of bucket distributed combine



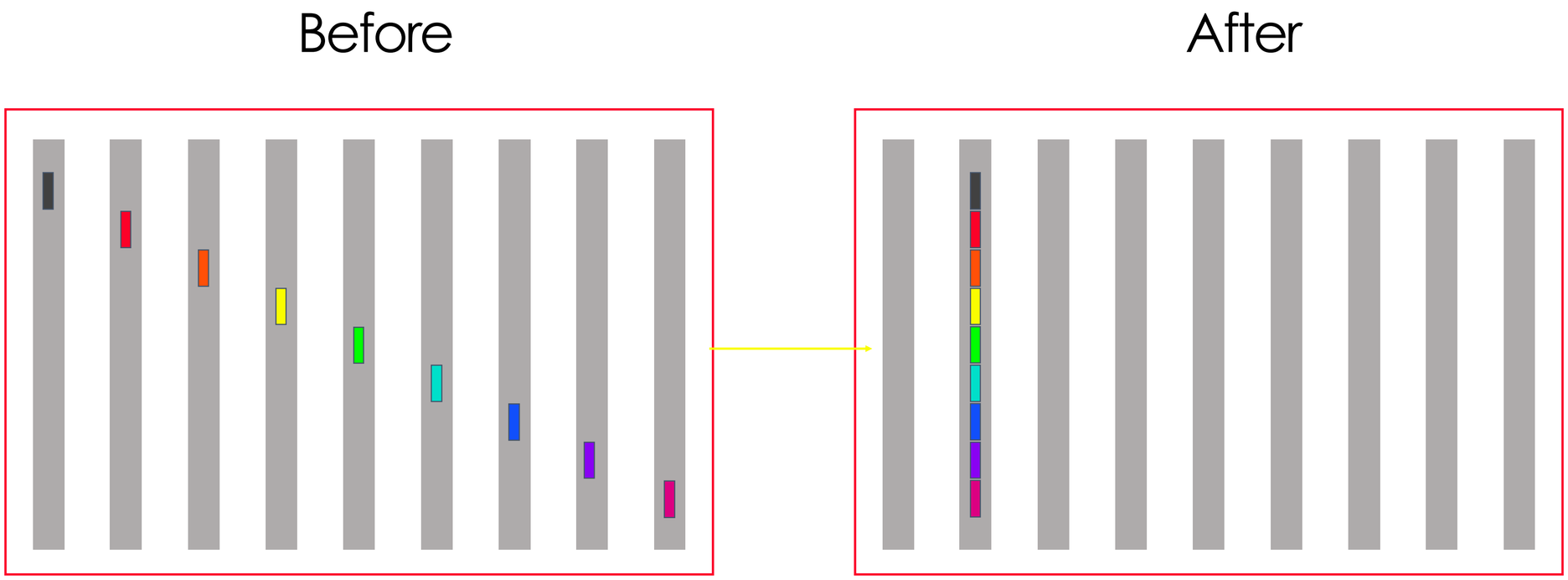
Scatter: Can Ring Be Better?

Notice: Scatter as implemented before using MST was optimal in Bandwidth as well (How to Prove?)



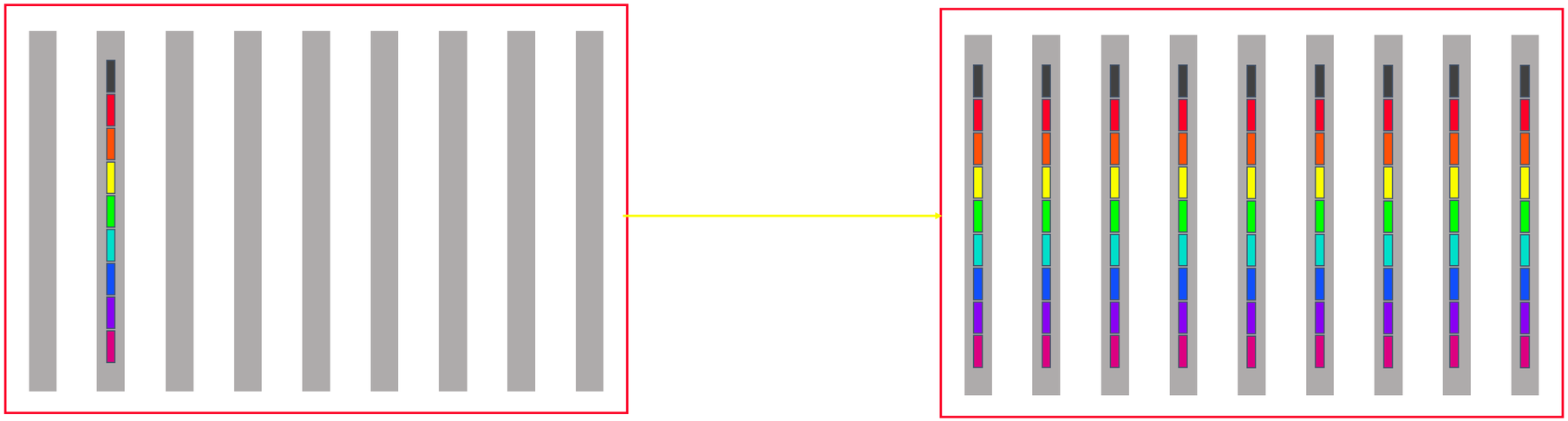
Gather

Notice: Gather as implemented before using MST was optimal in bandwidth as well (how to prove?)

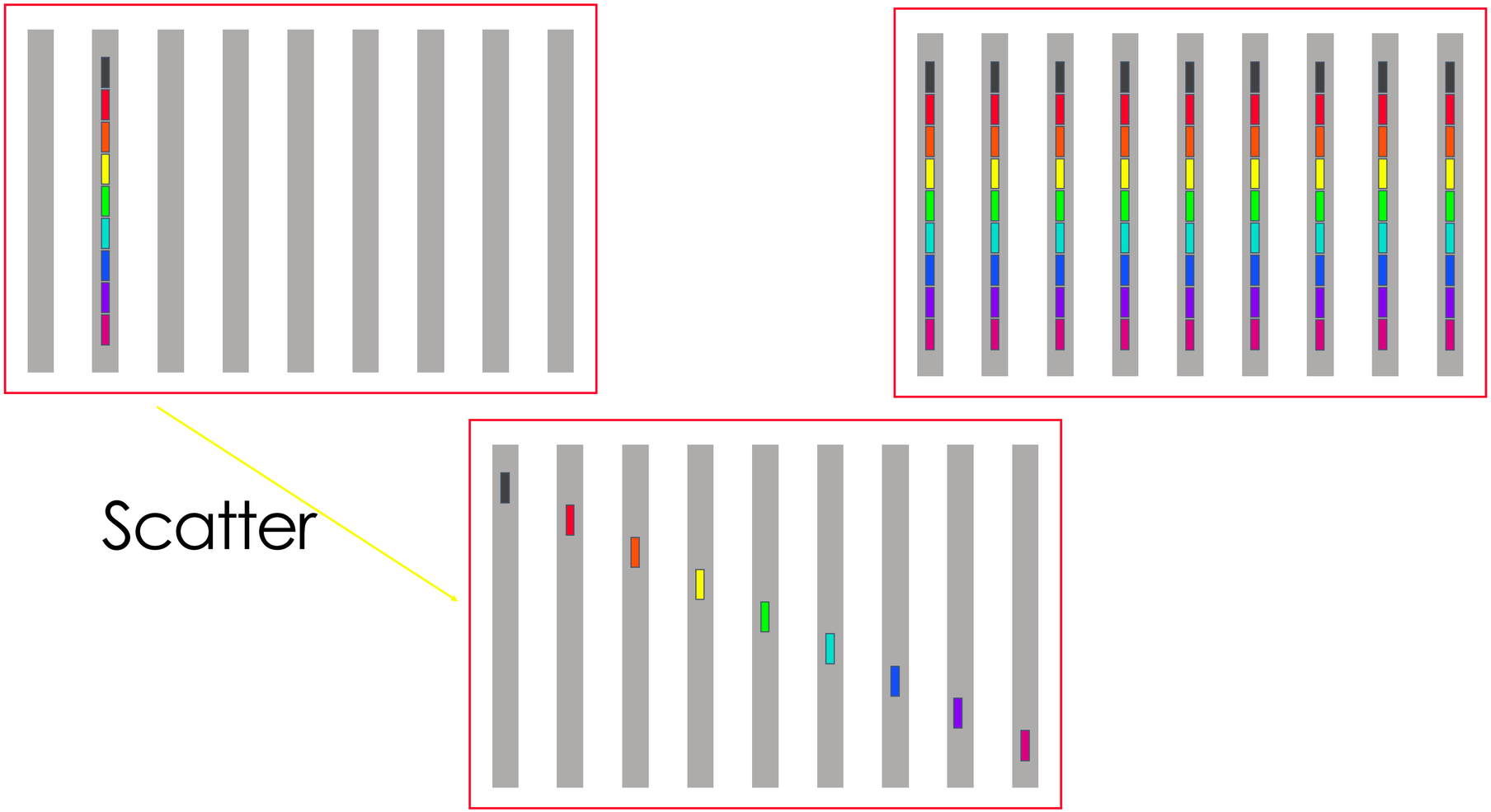


Using the building blocks

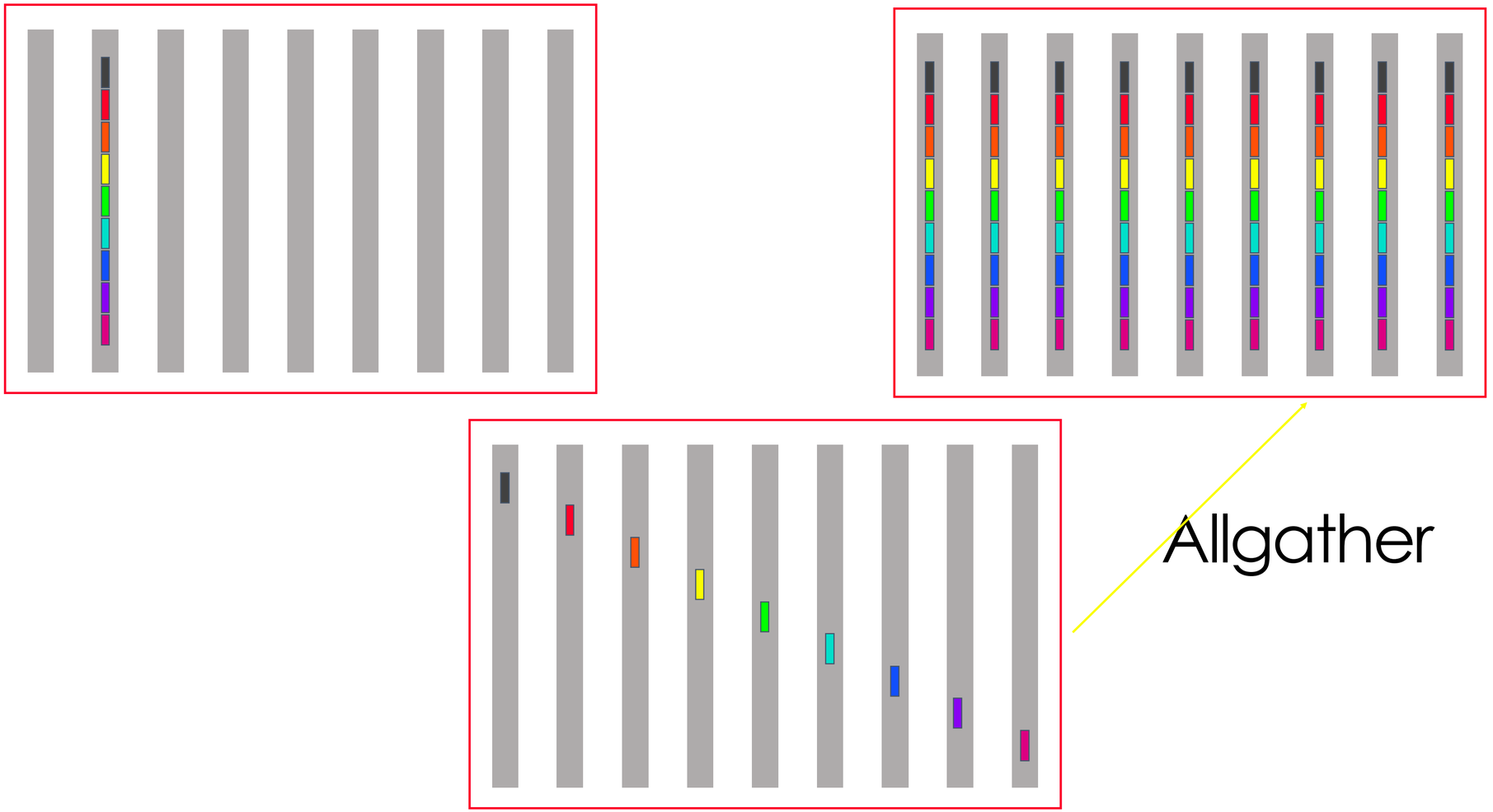
Broadcast (Large Message)



Broadcast (long vector)



Broadcast (long vector)



Cost of scatter/allgather broadcast

- Assumption: power of two number of nodes

scatter $$\log(p)\alpha + \frac{p-1}{p} n\beta$$

allgather $$(p-1)\alpha + \frac{p-1}{p} n\beta$$

$$\frac{(p-1)\alpha + \frac{p-1}{p} n\beta}{(\log(p) + p - 1)\alpha + 2\frac{p-1}{p} n\beta}$$

Cost of scatter/allgather broadcast

- Assumption: power of two number of nodes

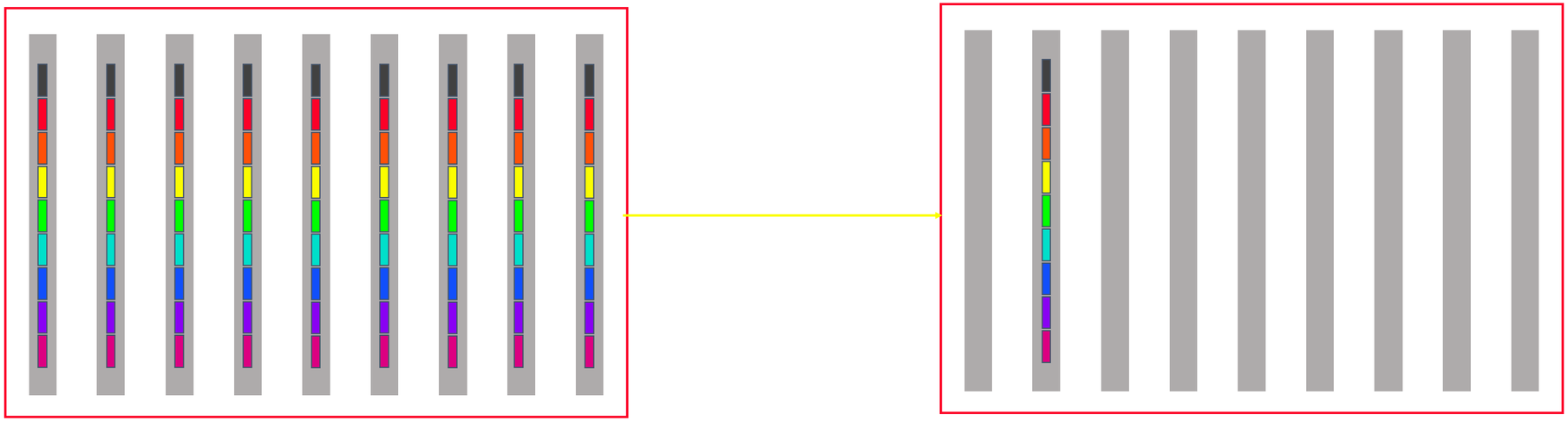
scatter $$\log(p)\alpha + \frac{p-1}{p} n\beta$$

allgather $$(p-1)\alpha + \frac{p-1}{p} n\beta$$

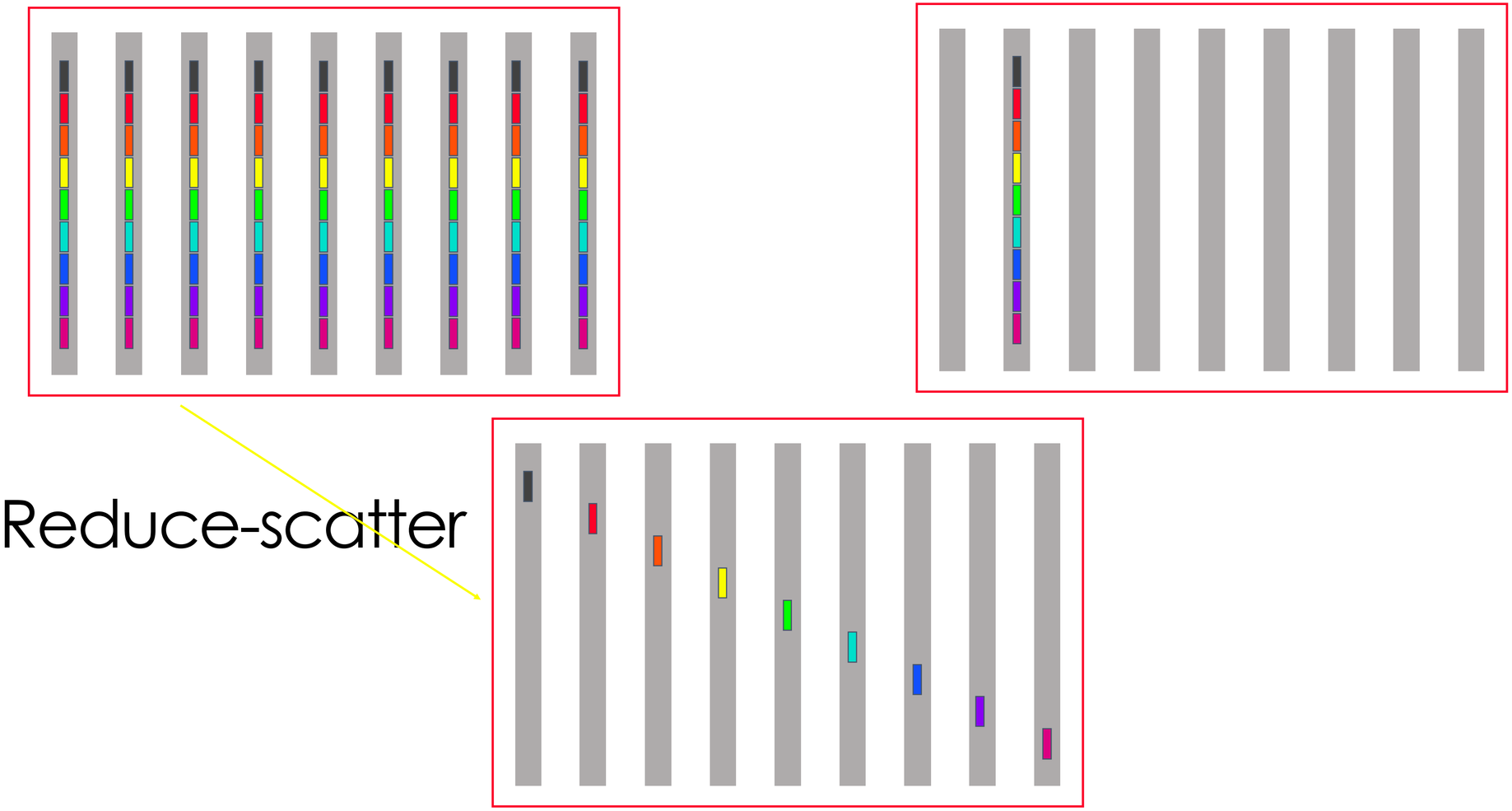
$$(\log(p) + p - 1)\alpha + 2\frac{p-1}{p} n\beta$$

Vs. MST broadcast: $$\lceil \log(p) \rceil \ (\alpha + n\beta)$$

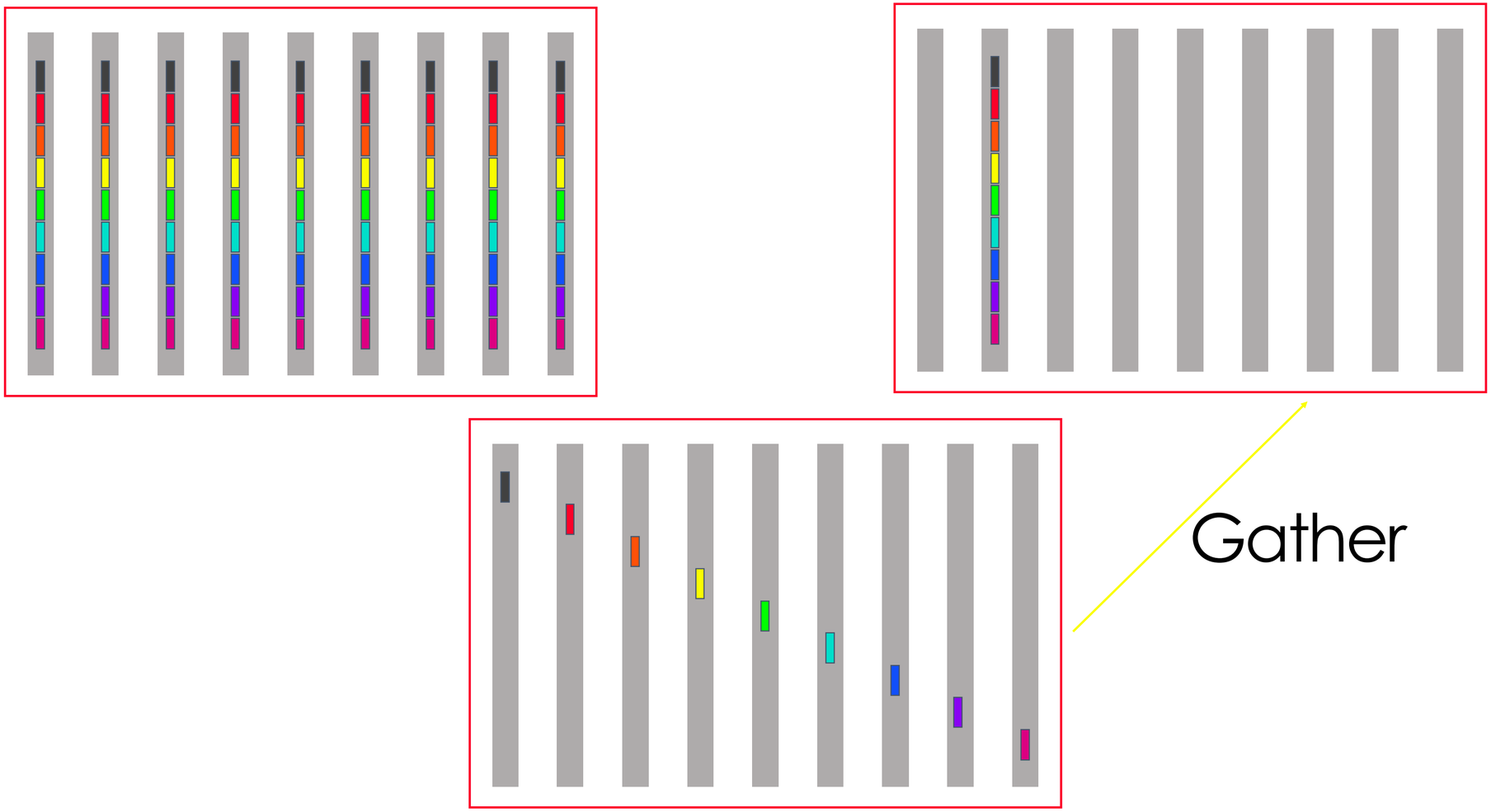
Reduce(-to-one) (long vector)



Reduce (long vector)



Combine-to-one (long vector)



Cost of Reduce-scatter/Gather Reduce(-to-one)

- Assumption: power of two number of nodes

Reduce-scatter $(p-1)\alpha + \frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$

gather $\log(p)\alpha + \frac{p-1}{p} n\beta$

$$\frac{\log(p)\alpha + \frac{p-1}{p} n\beta}{(\log(p) + p - 1)\alpha + 2\frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma}$$

Cost of Reduce-scatter/Gather Reduce(-to-one)

- Assumption: power of two number of nodes

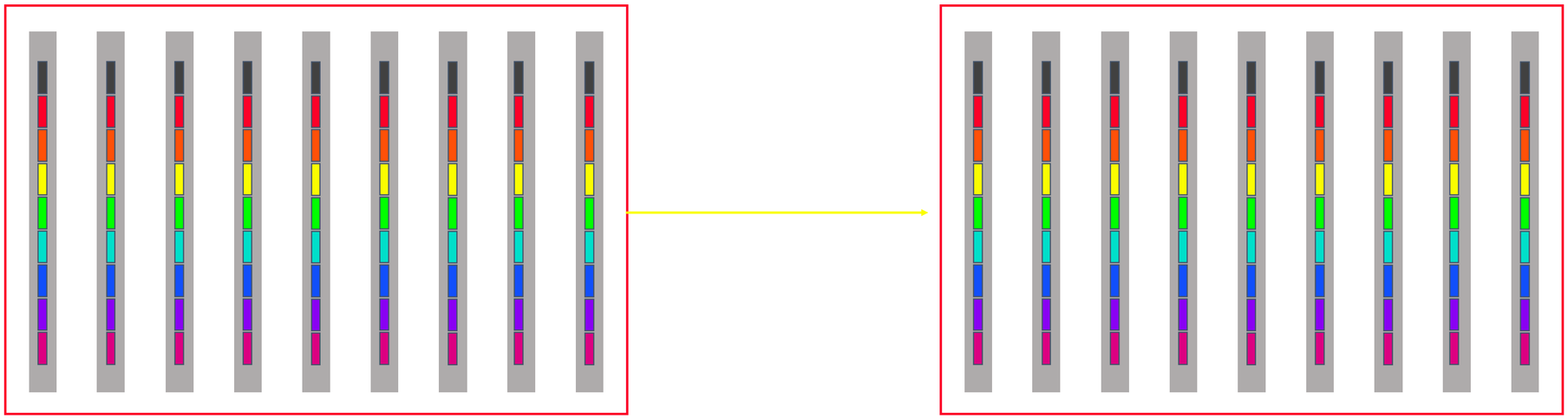
Reduce-scatter $(p-1)\alpha + \frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$

gather $\log(p)\alpha + \frac{p-1}{p} n\beta$

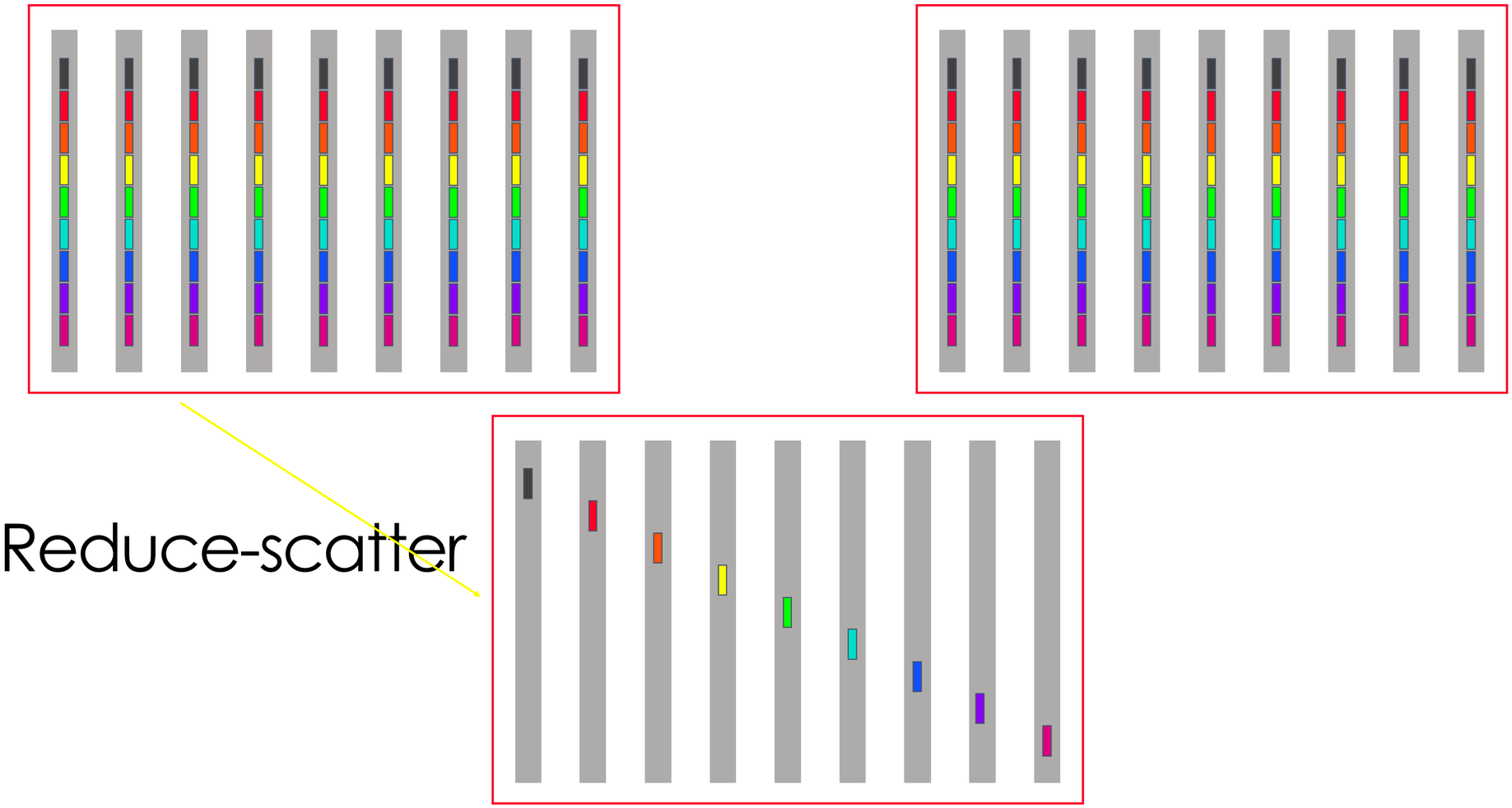
$$\frac{\log(p)\alpha + \frac{p-1}{p} n\beta}{(\log(p) + p - 1)\alpha + 2\frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma}$$

Vs. MST reduce: $\lceil \log(p) \rceil \ (\alpha + n\beta + n\gamma)$

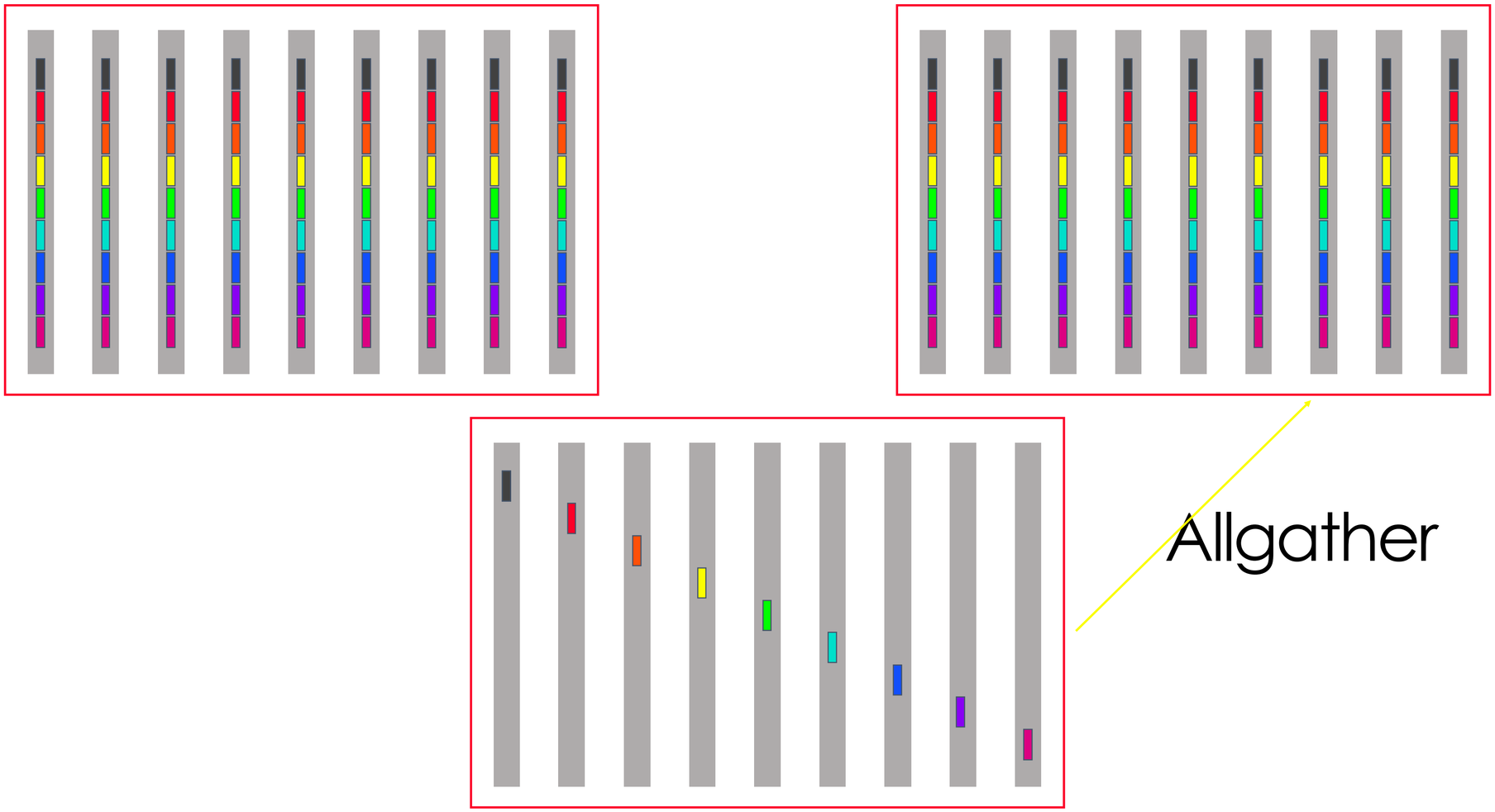
Allreduce (Large Message)



Allreduce (Large Message)



Allreduce
(long vector)



Cost of Reduce-scatter/Allgather Allreduce

- Assumption: power of two number of nodes

Reduce-scatter $(p-1)\alpha + \frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$

Allgather $(p-1)\alpha + \frac{p-1}{p} n\beta$

$$\frac{(p-1)\alpha + \frac{p-1}{p} n\beta}{2(p-1)\alpha + 2\frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma}$$

Cost of Reduce-scatter/Allgather Allreduce

- Assumption: power of two number of nodes

Reduce-scatter $(p-1)\alpha + \frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma$

Allgather $(p-1)\alpha + \frac{p-1}{p} n\beta$

$$\frac{(p-1)\alpha + \frac{p-1}{p} n\beta}{2(p-1)\alpha + 2\frac{p-1}{p} n\beta + \frac{p-1}{p} n\gamma}$$

Vs. Reduce-broadcast allreduce $2\log(p)\alpha + 2\log(p)n\beta + \log(p)n\gamma$

Recap

Reduce-scatter

$$(p-1)\alpha+\frac{p-1}{p}n(\beta+\gamma)$$

Scatter

$$log(p)\alpha+\frac{p-1}{p}n\beta$$

Gather

$$log(p)\alpha+\frac{p-1}{p}n\beta$$

Allgather

$$(p-1)\alpha+\frac{p-1}{p}n\beta$$

Reduce(-to-one)

Allreduce

Broadcast

Recap

Reduce-scatter

$(p-1)\alpha+\frac{p-1}{p}n(\beta+\gamma)$

Scatter

$log(p)\alpha+\frac{p-1}{p}n\beta$

Gather

$log(p)\alpha+\frac{p-1}{p}n\beta$

Allgather

$(p-1)\alpha+\frac{p-1}{p}n\beta$

Reduce(-to-one)

$(p-1+log(p))\alpha+\frac{p-1}{p}n(2\beta+\gamma)$

Allreduce

Broadcast

Recap

Reduce-scatter

$$(p-1)\alpha+\frac{p-1}{p}n(\beta+\gamma)$$

Scatter

$$\log(p)\alpha+\frac{p-1}{p}n\beta$$

Gather

$$\log(p)\alpha+\frac{p-1}{p}n\beta$$

Allgather

$$(p-1)\alpha+\frac{p-1}{p}n\beta$$

Reduce(-to-one)

$$(p-1+\log(p))\alpha+\frac{p-1}{p}n(2\beta+\gamma)$$

Allreduce

$$2(p-1)\alpha+\frac{p-1}{p}n(2\beta+\gamma)$$

Broadcast

$$(\log(p)+p-1)\alpha+2\frac{p-1}{p}n\beta$$

Recap

Reduce-scatter

$$(p-1)\alpha+\frac{p-1}{p}n(\beta+\gamma)$$

Scatter

$$\log(p)\alpha+\frac{p-1}{p}n\beta$$

Gather

$$\log(p)\alpha+\frac{p-1}{p}n\beta$$

Allgather

$$(p-1)\alpha+\frac{p-1}{p}n\beta$$

Reduce(-to-one)

$$(p-1+\log(p))\alpha+\frac{p-1}{p}n(2\beta+\gamma)$$

Allreduce

$$2(p-1)\alpha+\frac{p-1}{p}n(2\beta+\gamma)$$

Broadcast

Recap

Reduce-scatter

$$(p-1)\alpha+\frac{p-1}{p}n(\beta+\gamma)$$

Scatter

$$\log(p)\alpha+\frac{p-1}{p}n\beta$$

Gather

$$\log(p)\alpha+\frac{p-1}{p}n\beta$$

Allgather

$$(p-1)\alpha+\frac{p-1}{p}n\beta$$

Reduce(-to-one)

$$(p-1+\log(p))\alpha+\frac{p-1}{p}n(2\beta+\gamma)$$

Allreduce

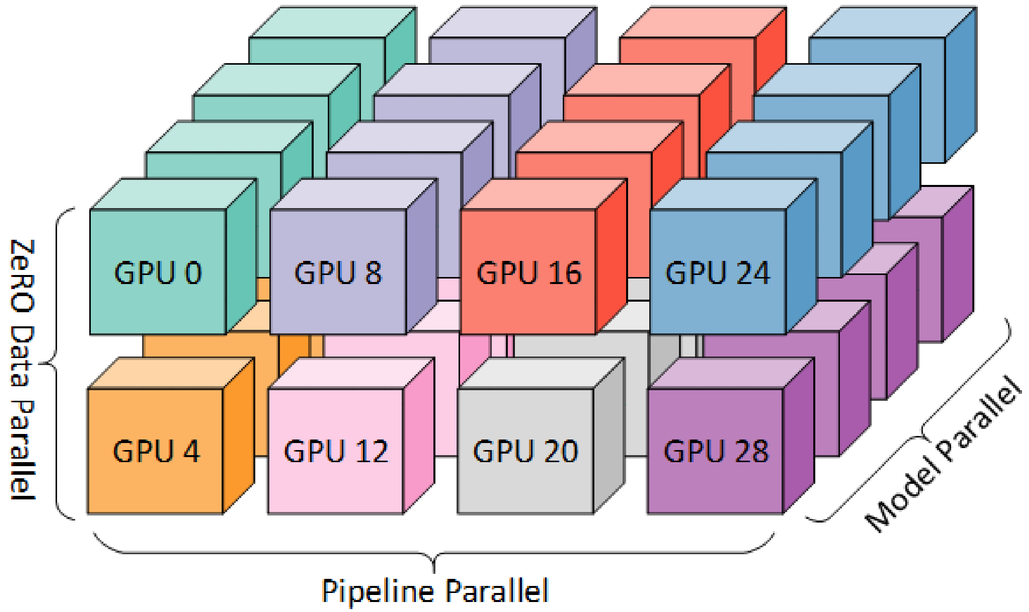
$$2(p-1)\alpha+\frac{p-1}{p}n(2\beta+\gamma)$$

Broadcast

$$(\log(p)+p-1)\alpha+2\frac{p-1}{p}n\beta$$

A More Complicate Case

- Real Cluster to train ChatGPT:
  - If using GPU: 2D Mesh
  - If using TPU: 3D Mesh, see figure below



Example: 2D Broadcast

- Idea: Use 1D to compose 2

Example: 2D Broadcast

- Idea: Use 1D to compose 2
- Option 1:
  - **MST broadcast in column**

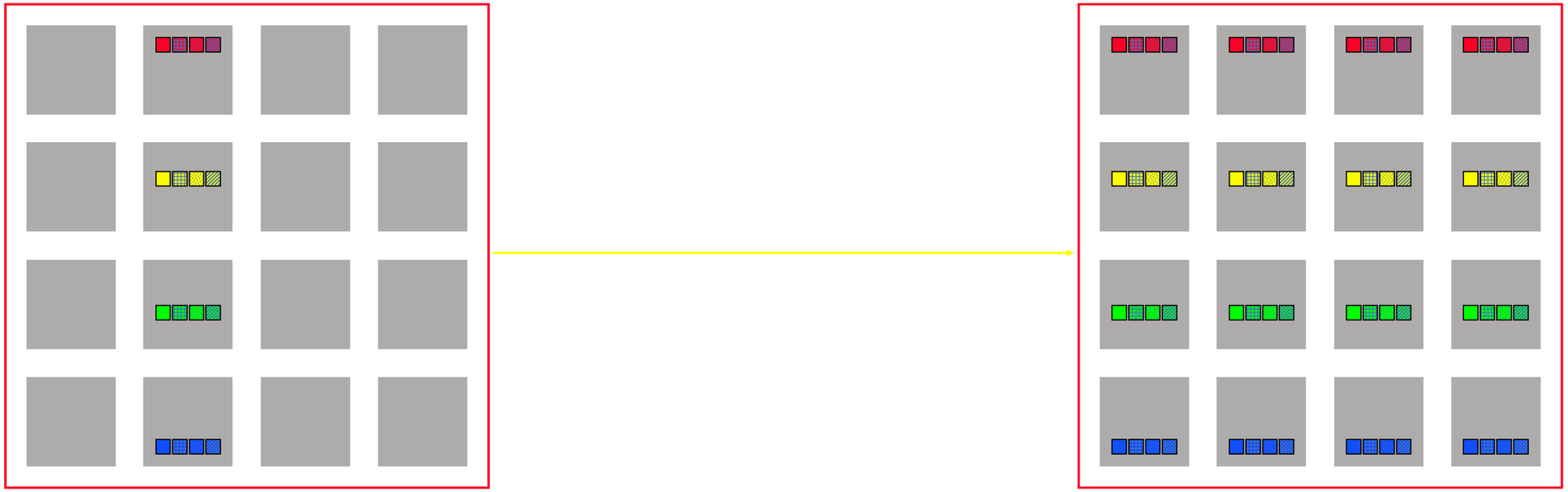
Example: 2D Broadcast

- Option 1:
  - MST broadcast in column
  - **MST broadcast in rows**

Example: 2D Broadcast

- Option 2:
  - Scatter in column

Example: 2D Broadcast



- Option 2:
  - Scatter in column
  - MST broadcast in rows

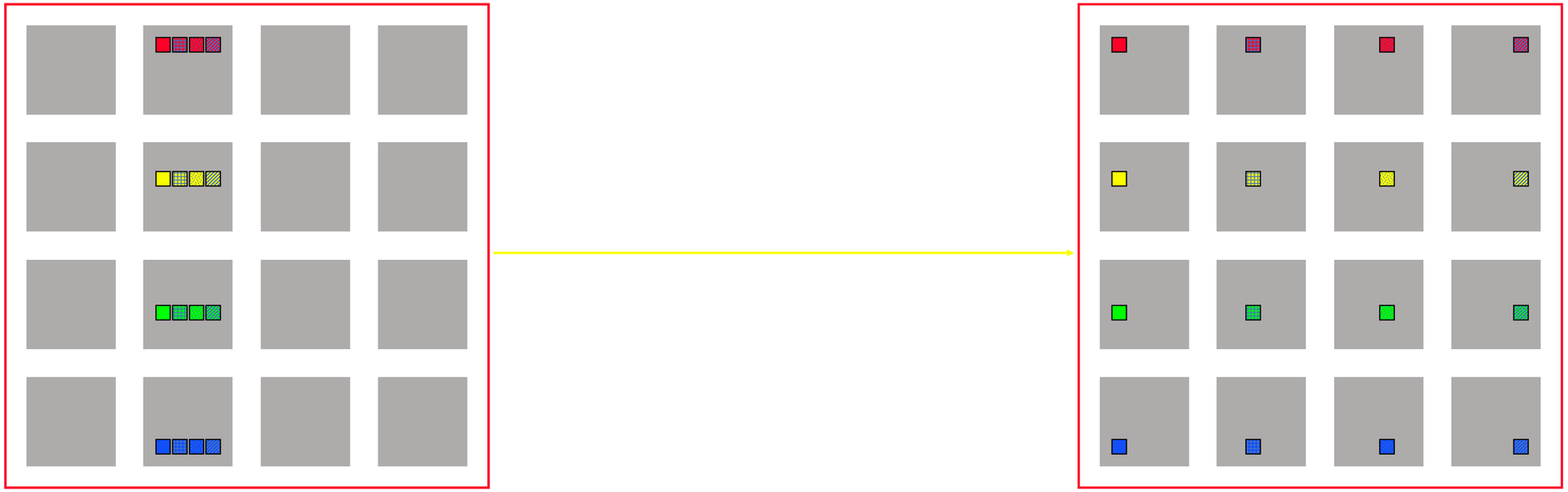
Example: 2D Broadcast

- Option 2:
  - Scatter in column
  - MST broadcast in rows
  - Allgather in columns

Example: 2D Broadcast

- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

Example: 2D Broadcast



- Option 3:
  - Scatter in column
  - Scatter in rows

Example: 2D Broadcast

- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows

Example: 2D Broadcast

- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

Cost comparison

- **Option 1:**
  - **MST broadcast in column**
  - **MST broadcast in rows**
- Option 2:
  - Scatter in column
  - MST broadcast in rows
  - Allgather in columns
- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

$$log(c)\alpha + log(c)n\beta$$

$$\frac{log(r)\alpha + log(r)n\beta}{log(p)\alpha + log(p)n\beta}$$

Cost comparison

- Option 1:
  - MST broadcast in column
  - MST broadcast in rows
- **Option 2:**
  - **Scatter in column**
  - **MST broadcast in rows**
  - **Allgather in columns**
- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

$$log(c)\alpha + \frac{c-1}{c} n\beta$$

$$log(r)\alpha + log(r)\frac{n}{c}\beta$$

$$\frac{(c-1)\alpha + \frac{c-1}{c} n\beta}{(log(p)+c-1)\alpha + \left(2\frac{c-1+log(r)}{c}\right)n\beta}$$

Cost comparison

- Option 1:
  - MST broadcast in column
  - MST broadcast in rows
- Option 2:
  - Scatter in column
  - MST broadcast in rows
  - Allgather in columns
- **Option 3:**
  - **Scatter in column**
  - **Scatter in rows**
  - **Allgather in rows**
  - **Allgather in columns**

$$log(c)\alpha + \frac{c-1}{c} n\beta$$

$$log(r)\alpha + \frac{r-1}{r}\frac{n}{c}\beta$$

$$(r-1)\alpha + \frac{r-1}{r}\frac{n}{c}\beta$$

$$(c-1)\alpha + \frac{c-1}{c} n\beta$$

$$\overline{(log(p) + r + c - 2)\alpha + 2\frac{p-1}{p} n\beta}$$

Cost comparison

- Option 1:
  - MST broadcast in column
  - MST broadcast in rows
- Option 2:
  - Scatter in column
  - MST broadcast in rows
  - Allgather in columns
- Option 3:
  - Scatter in column
  - Scatter in rows
  - Allgather in rows
  - Allgather in columns

$$log(p)\alpha + log(p)n\beta$$

$$(log(p) + c - 1)\alpha + \left(2\frac{c - 1 + log(r)}{c}\right)n\beta$$

$$(log(p) + r + c - 2)\alpha + 2\frac{p-1}{p}n\beta$$

Summary and Question

- MST -> when alpha dominates
- Ring -> when n*beta dominates
- 2D can be composed using 1D, 3D can be composed using 2D, …
- Latency / Bandwidth trade-offs

Recap

- Q1: Which collective primitive maps to the distributed SGD gradient synchronization step?
- Q2: How many messages do we need to transfer over the network for a single iteration of GPT-3 SGD update assuming 8-gpu parallelism?
- Q3: For Q2, assuming 1D mesh, should we use MST or Ring?

Collective Pros

- A set of structured / well-defined communication primitives
- Extremely well-optimized
- Beautiful math, easy to analyze, and easy to understand its performance

Collective Cons

- Lack of Fault Tolerance
  - What if one node (in the ring) is dead?
- Requires Homogeneity
  - What if one node computes slower than all other nodes?
  - What if one link has lower bandwidth than the other node?

Real Cluster:

- Need Fault tolerance
- Heterogeneous hardware setup

Next Topics

- This week 2 classes: Data base + Cloud Storage
  - Delta from previous year offering:
    - we skip a substantial part of relational database
    - spend more time one networking, HPC, and ML

- Next week: Parallelism and Big Data processing
  - We will come back to study how we address the problem of Collectives