Algorithms: Greedy Method

An Activity-Selection Problem

Dr. Md. Abul Kashem Mia, Professor, CSE Dept, BUET

Greedy Algorithms: Principles

- A *greedy algorithm* always makes the choice that looks best at the moment.
- A greedy algorithm works in phases.

  At each phase:
  - You take the best you can get right now, without regard for future consequences.
  - You hope that by choosing a local optimum at each step, you will end up at a global optimum.
  - For some problems, it works.

An Activity Selection Problem

- Input: A set of activities S = $\{a_1, \ldots, a_n\}$
  - Each activity a_i has a start time s_i and a finish time f_i, where $0 \leq s_i < f_i < \infty$
  - If selected, activity a_i takes place during the half-open time interval $[s_i, f_i)$
- Two activities are compatible if and only if their intervals do not overlap
- Output: a maximum-size subset of mutually compatible activities

The Activity Selection Problem

- Here are a set of start and finish times

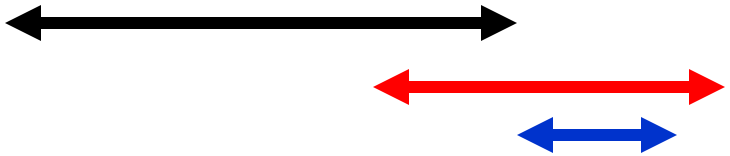
i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	8	9	10	11	12	13	14

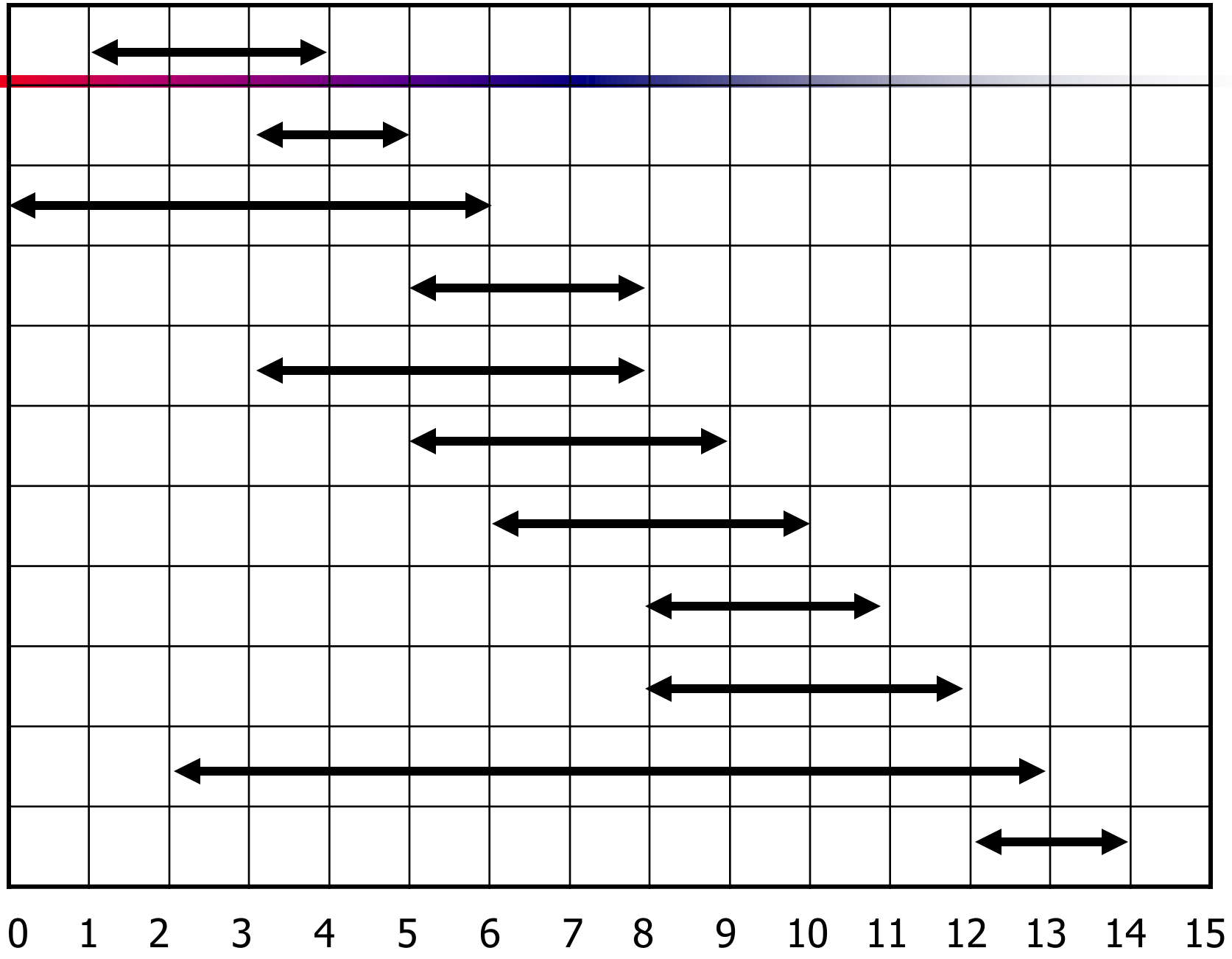
- What is the maximum number of activities that can be completed?
  - $\{a_3, a_9, a_{11}\}$ can be completed
  - But so can $\{a_1, a_4, a_8, a_{11}\}$ which is a larger set
  - But it is not unique, consider $\{a_2, a_4, a_9, a_{11}\}$

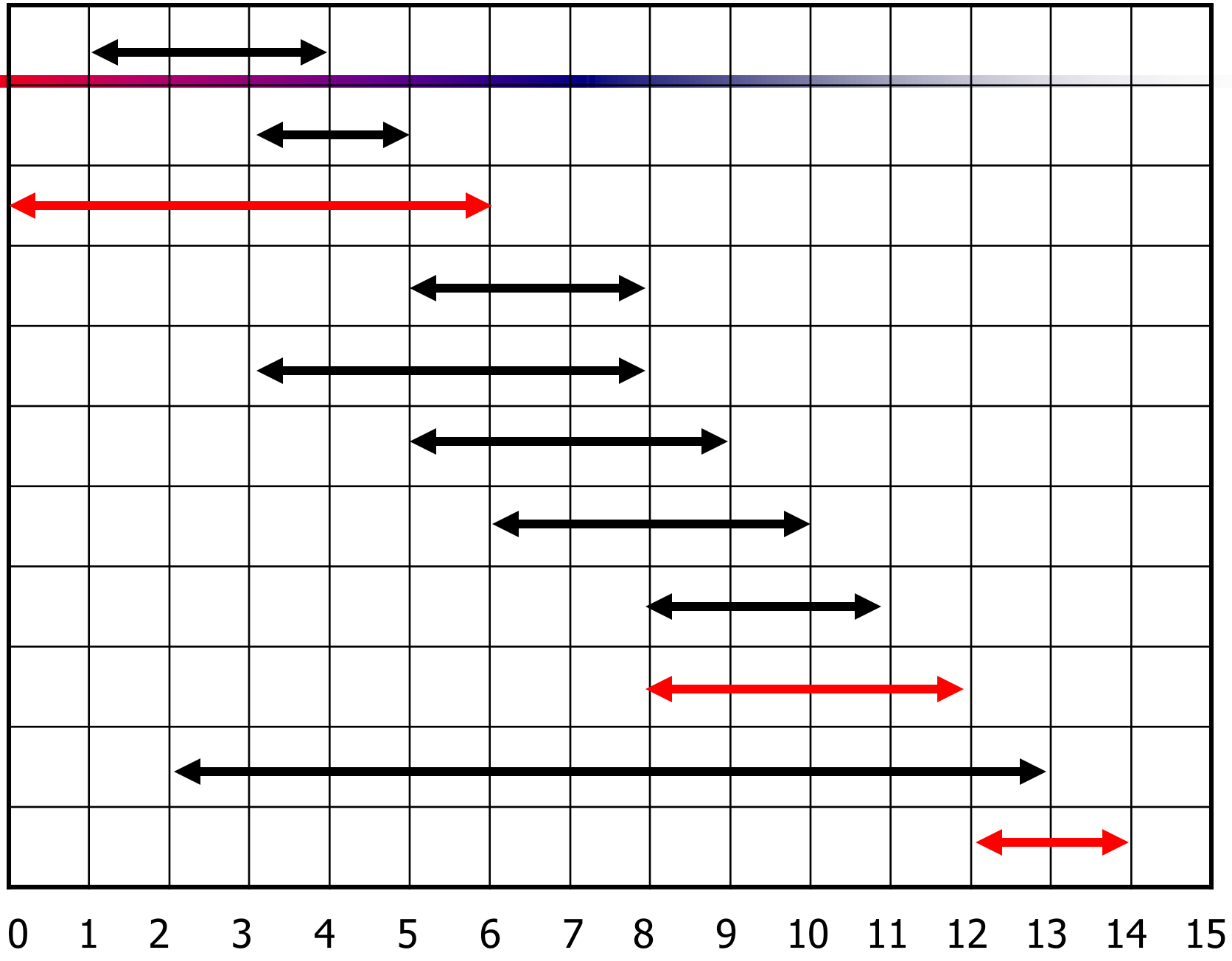
Interval Representation

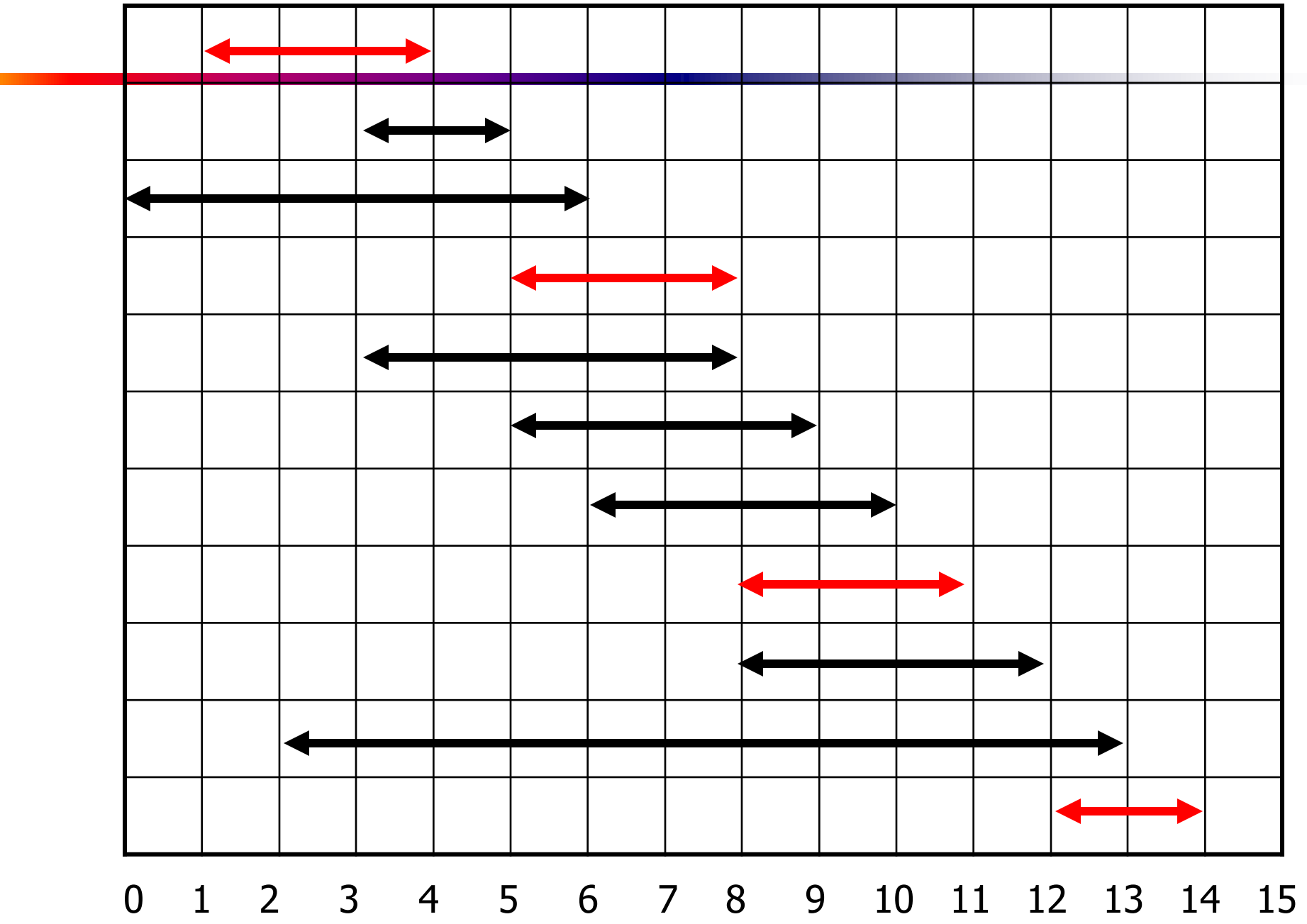
- Here are a set of start and finish times

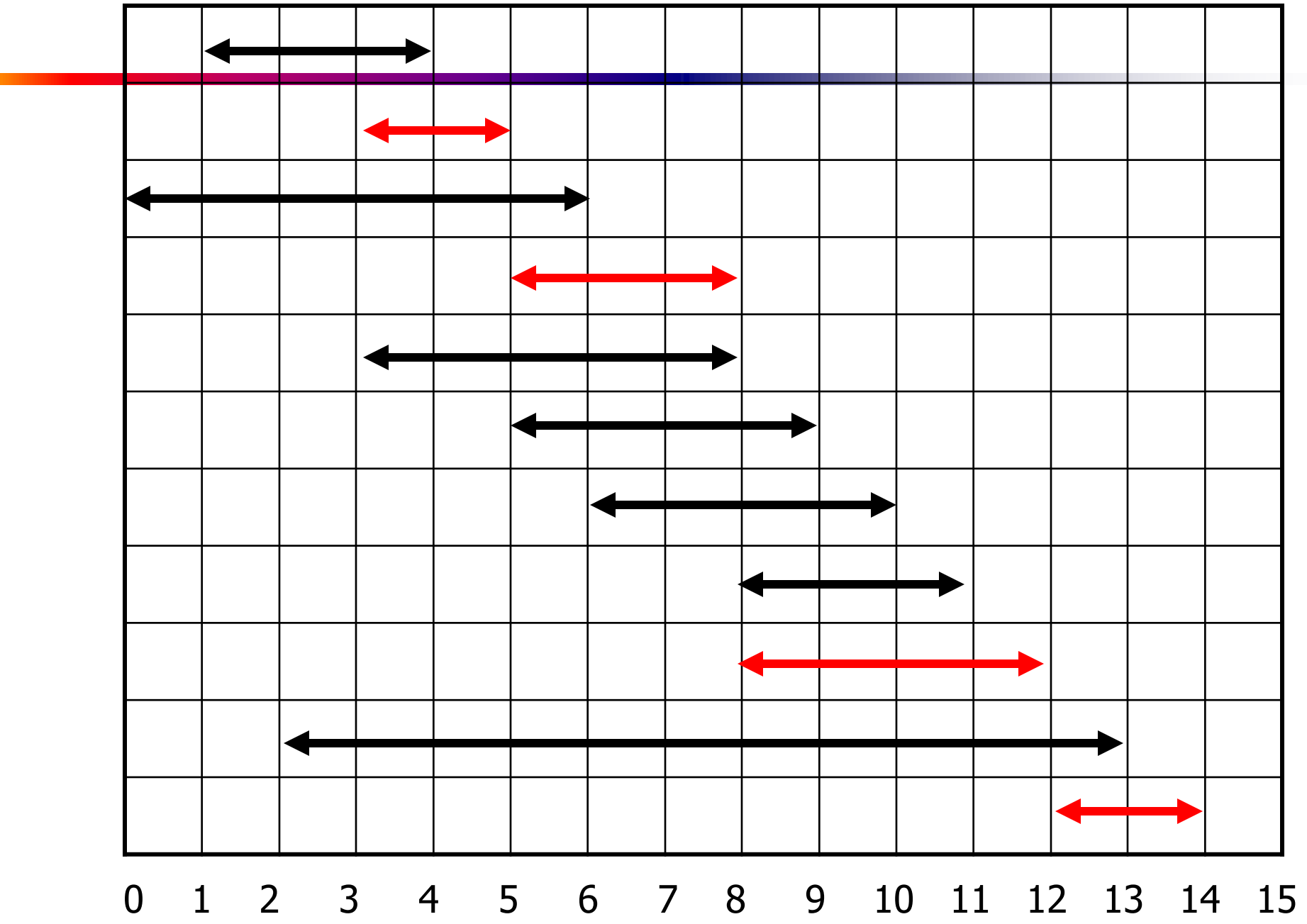
i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	8	9	10	11	12	13	14




0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

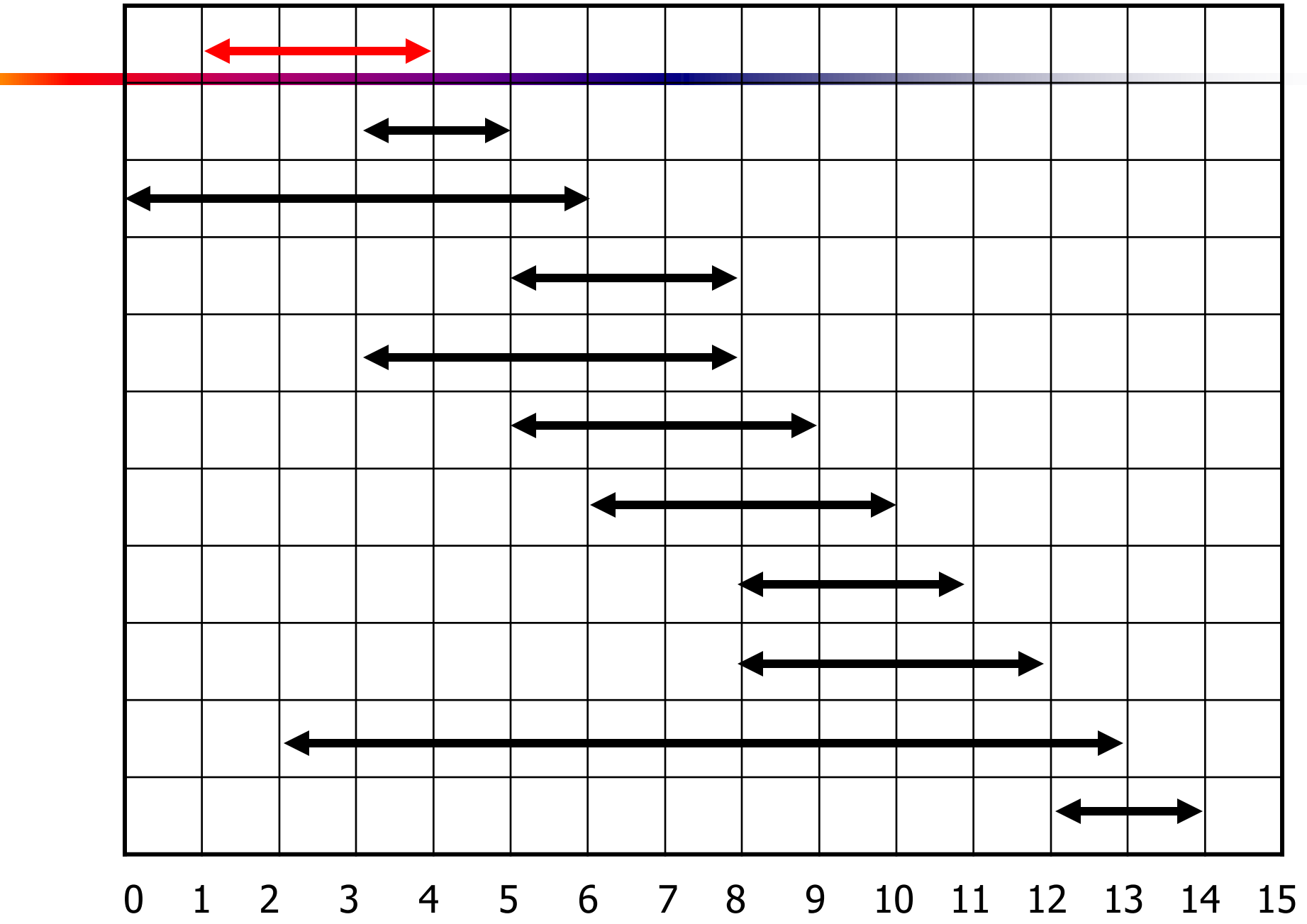

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

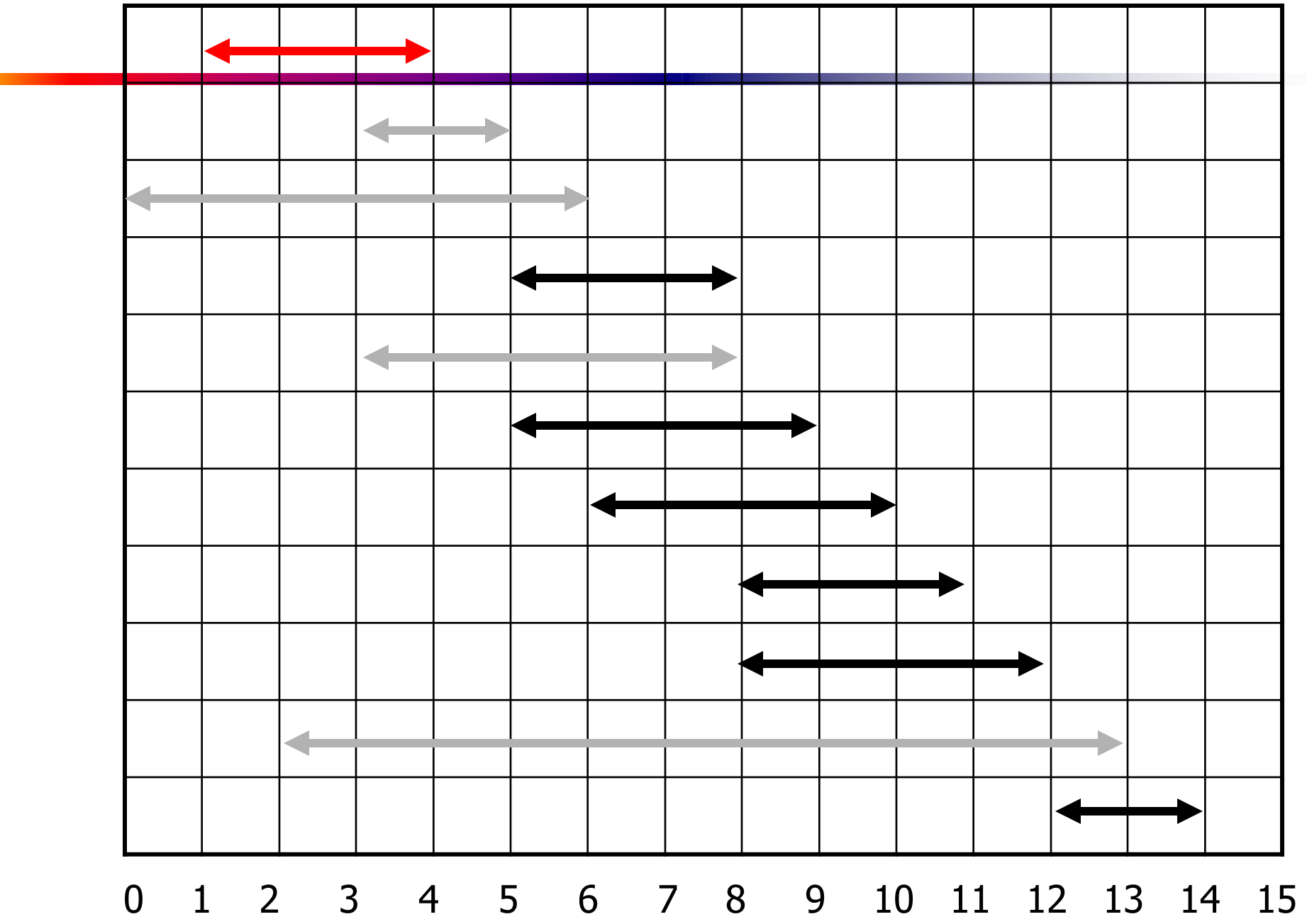

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

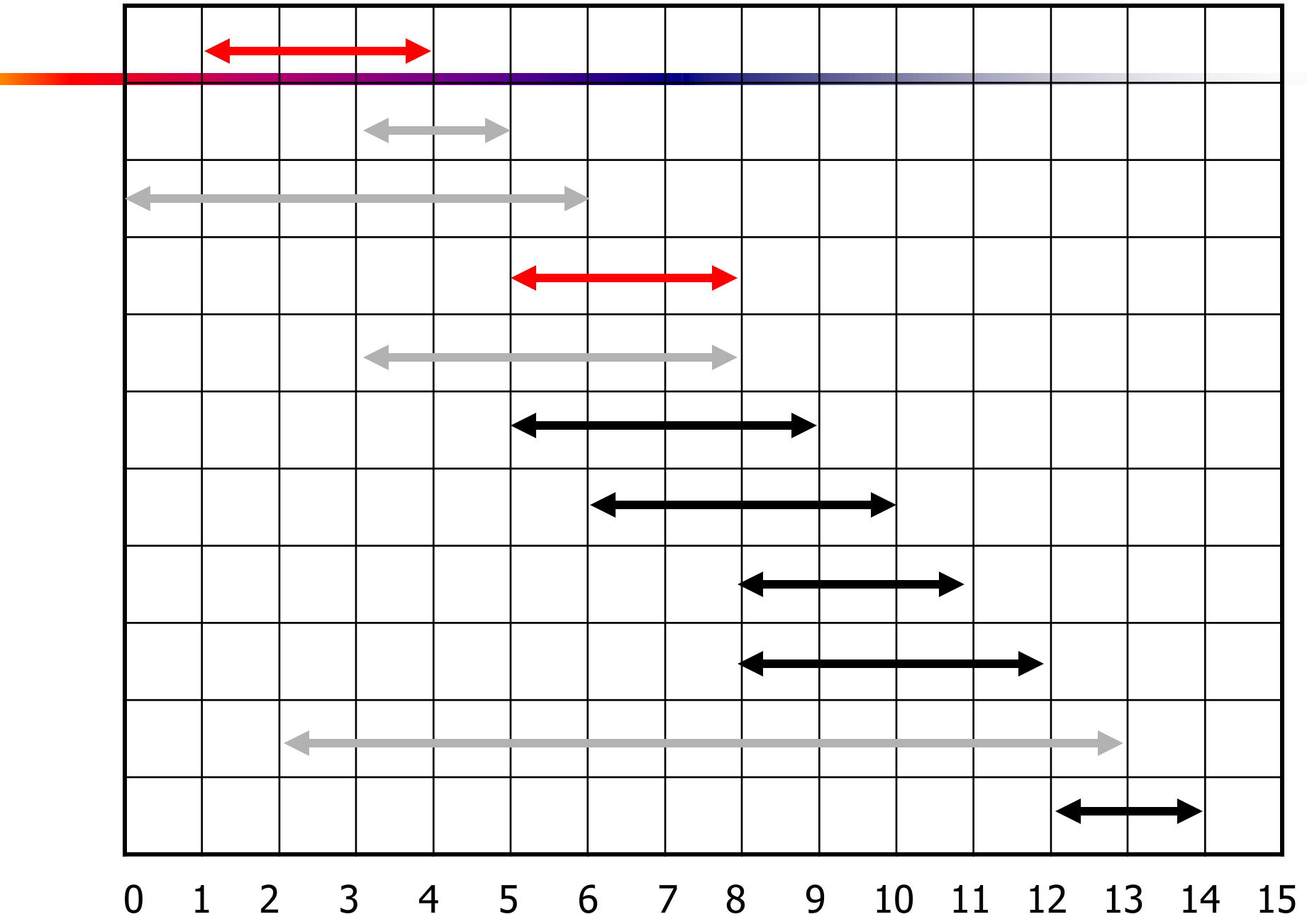

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

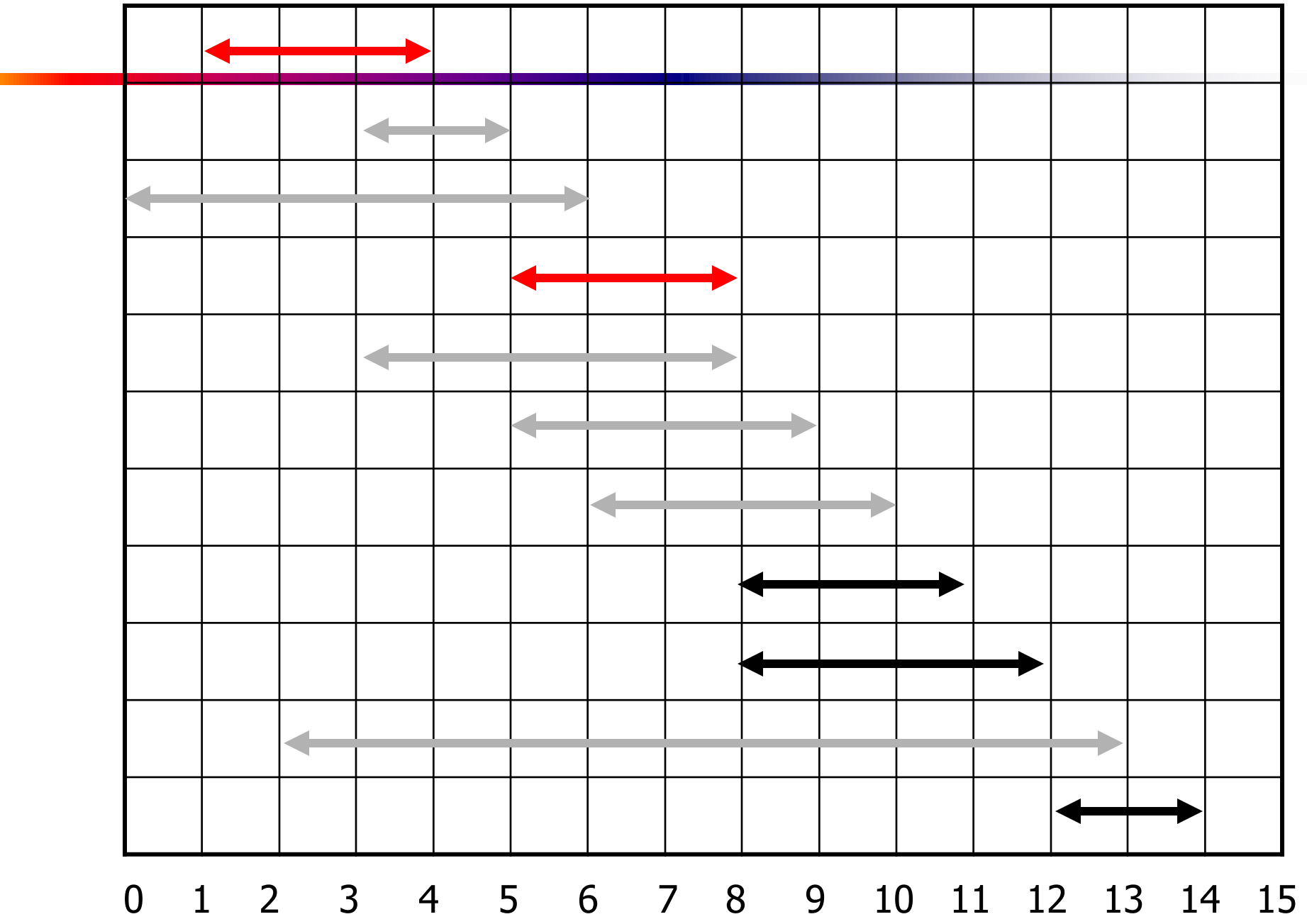
Early Finish Greedy

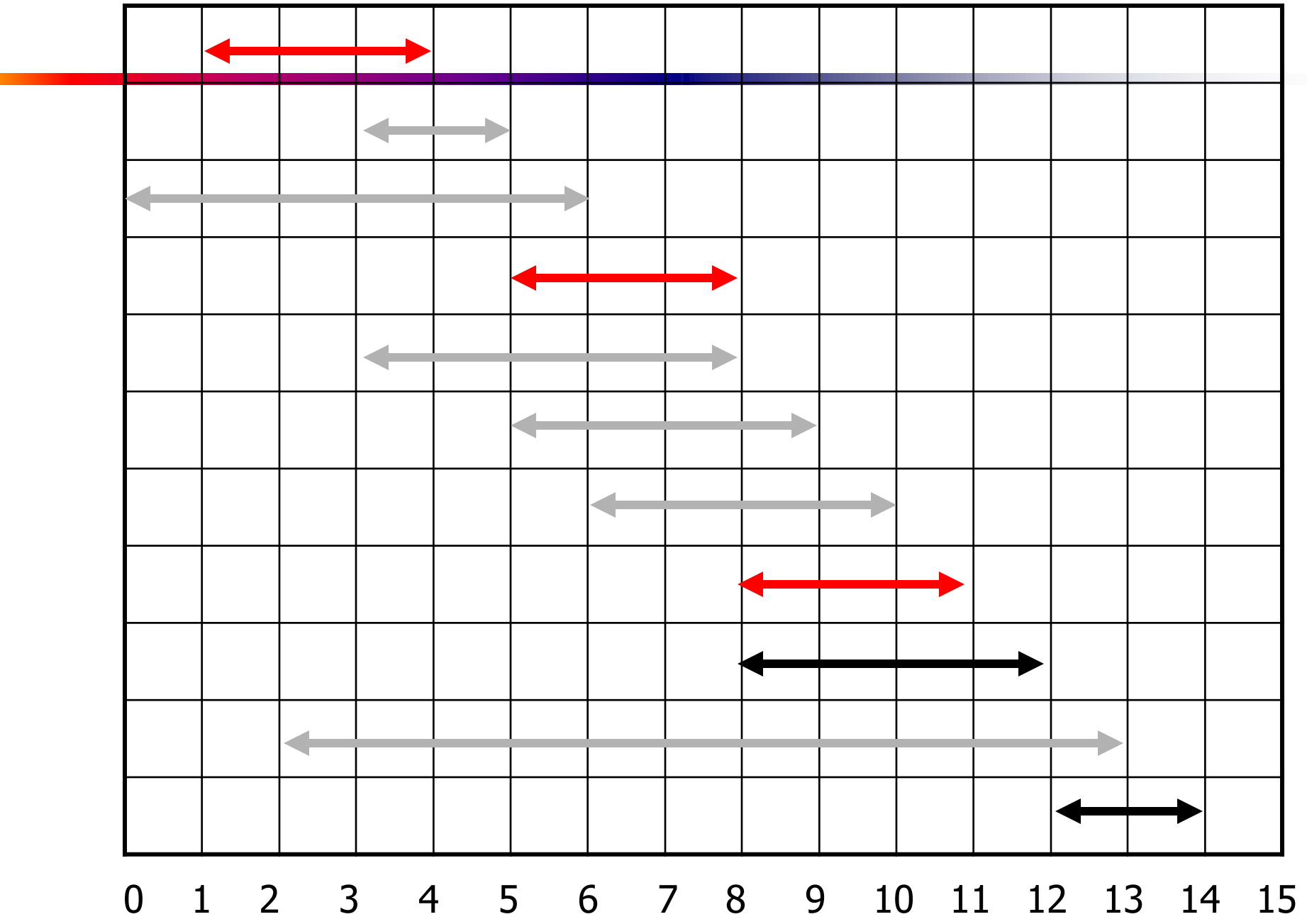
- Select the activity with the earliest finish
- Eliminate the activities that could not be scheduled
- Repeat!

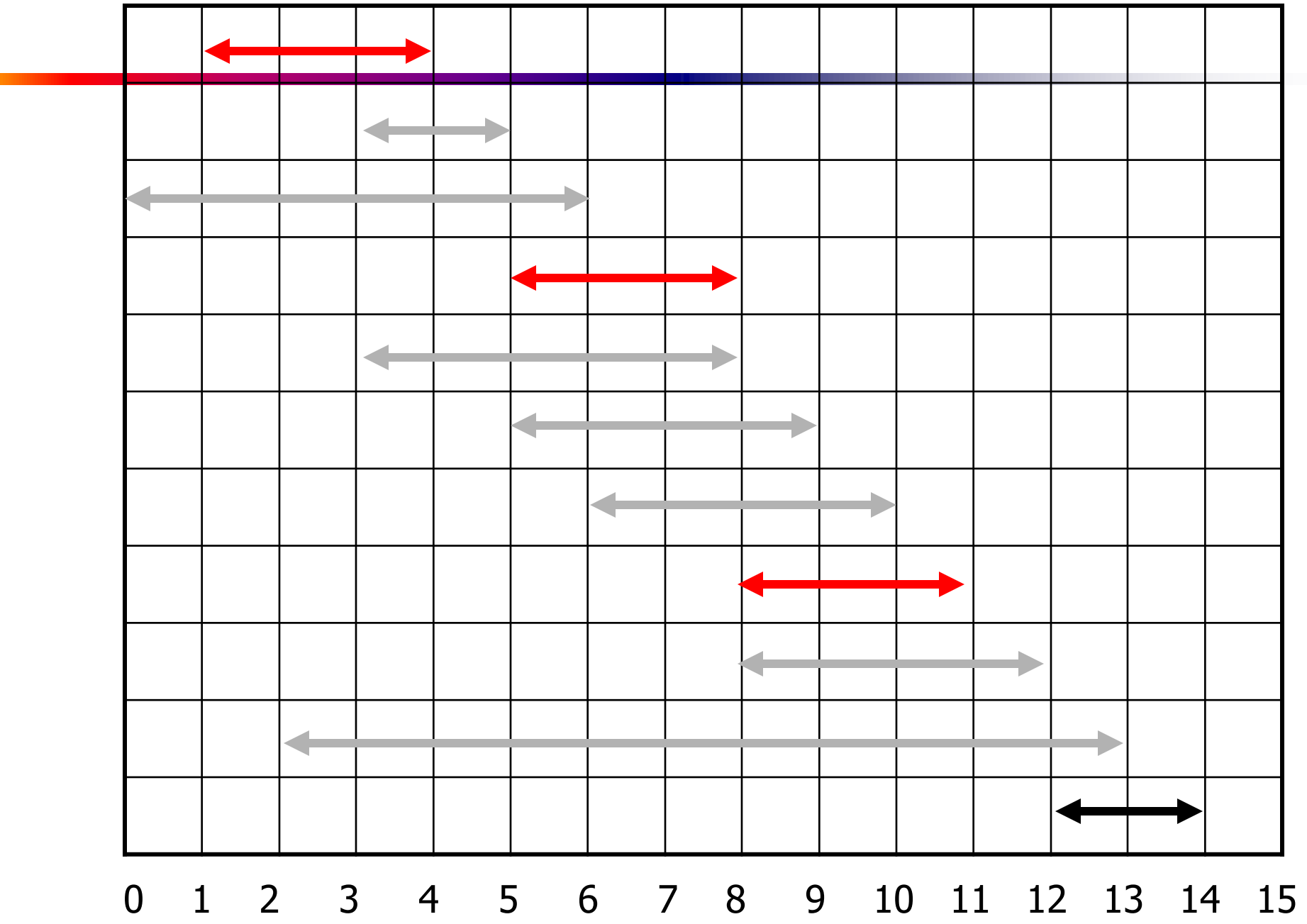

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

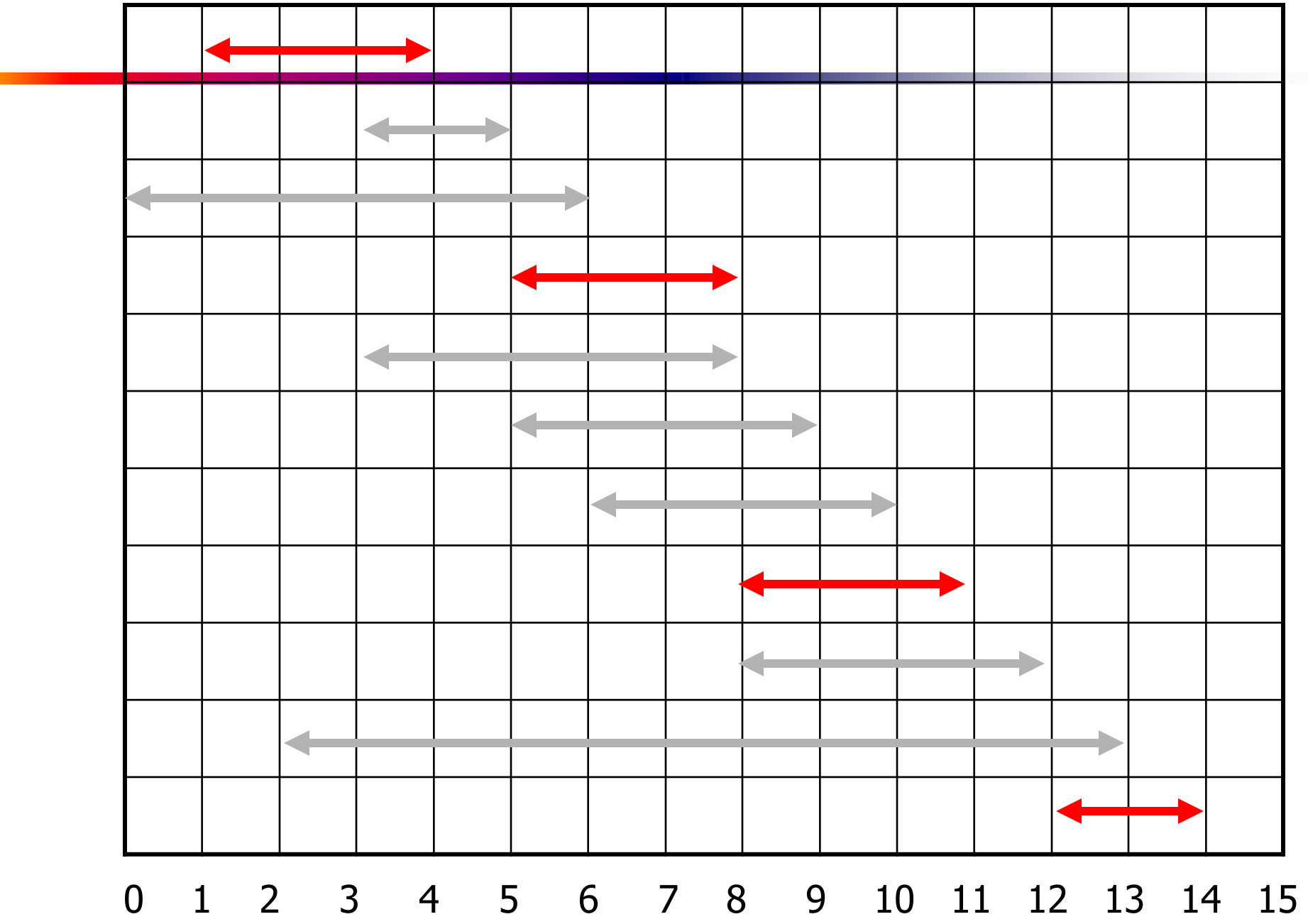

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Assuming activities are sorted by finish time

```
GREEDY-ACTIVITY-SELECTOR(s, f)
1   n ← length[s]
2   A ← {a_1}
3   i ← 1
4   for m ← 2 to n
5         do if s_m ≥ f_i
6               then A ← A ∪ {a_m}
7                    i ← m
8   return A
```