Activity Selection Problem
Book: Cormen; Section: 16.1

An Activity Selection Problem

- **Input:** A set of activities S = $\{a_1, \ldots, a_n\}$
- Each activity has start time and a finish time
  - $a_i = (s_i, f_i)$
- Two activities(a_i, a_j) are **compatible** if and only if their interval does not overlap
  - $s_i \geq f_j$ **OR** $s_j \geq f_i$
- **Output:** a maximum-size subset of mutually compatible activities

Example (Page - 415)

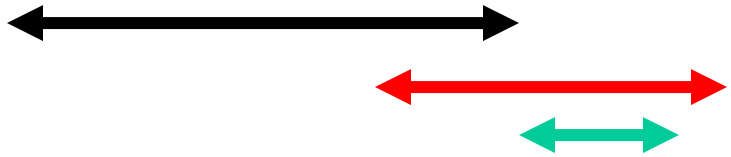
- Here are a set of start and finish times

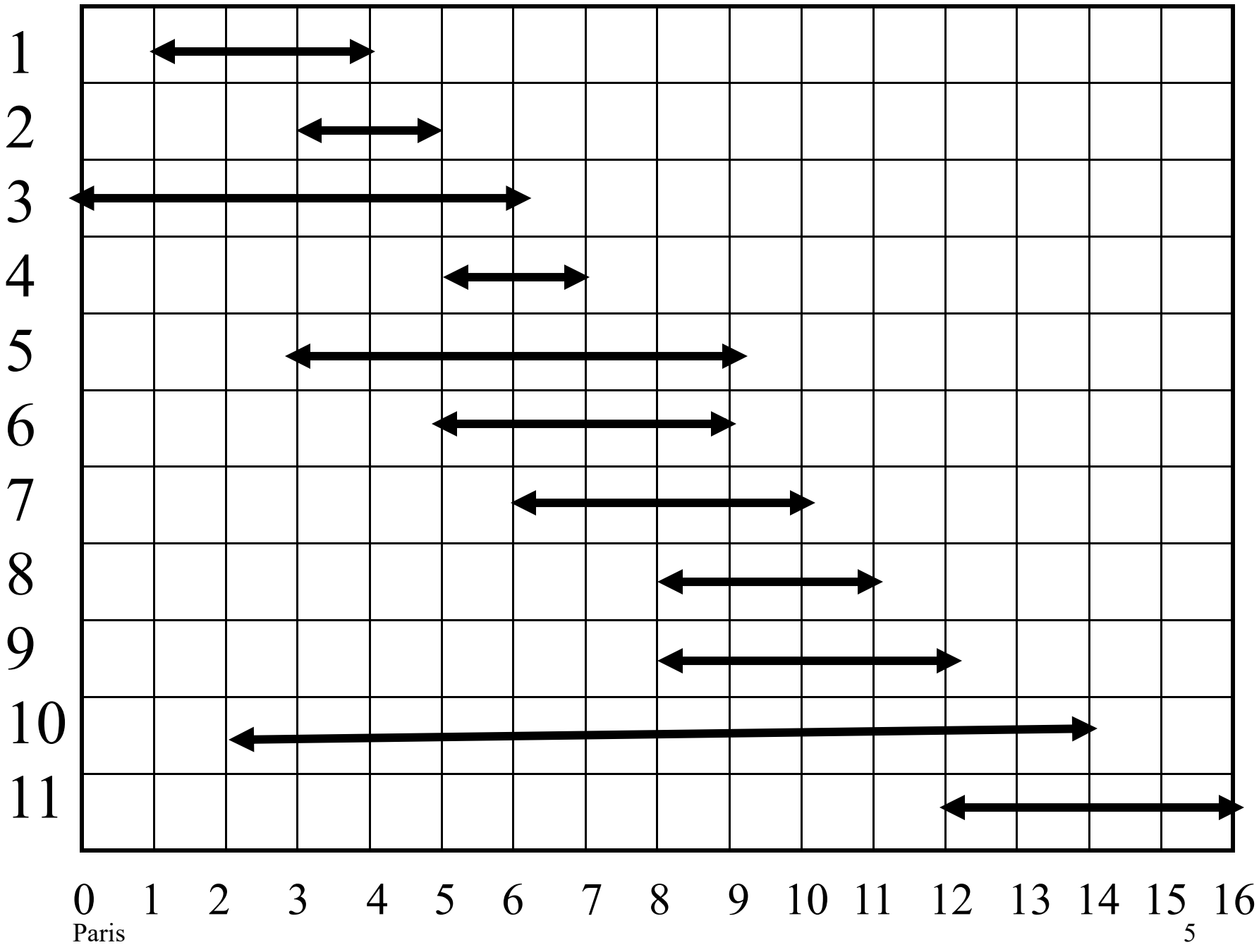
Activity No (i)	1	2	3	4	5	6	7	8	9	10	11
Start time (s_i)	1	3	0	5	3	5	6	8	8	2	12
Finish time (f_i)	4	5	6	7	9	9	10	11	12	14	16

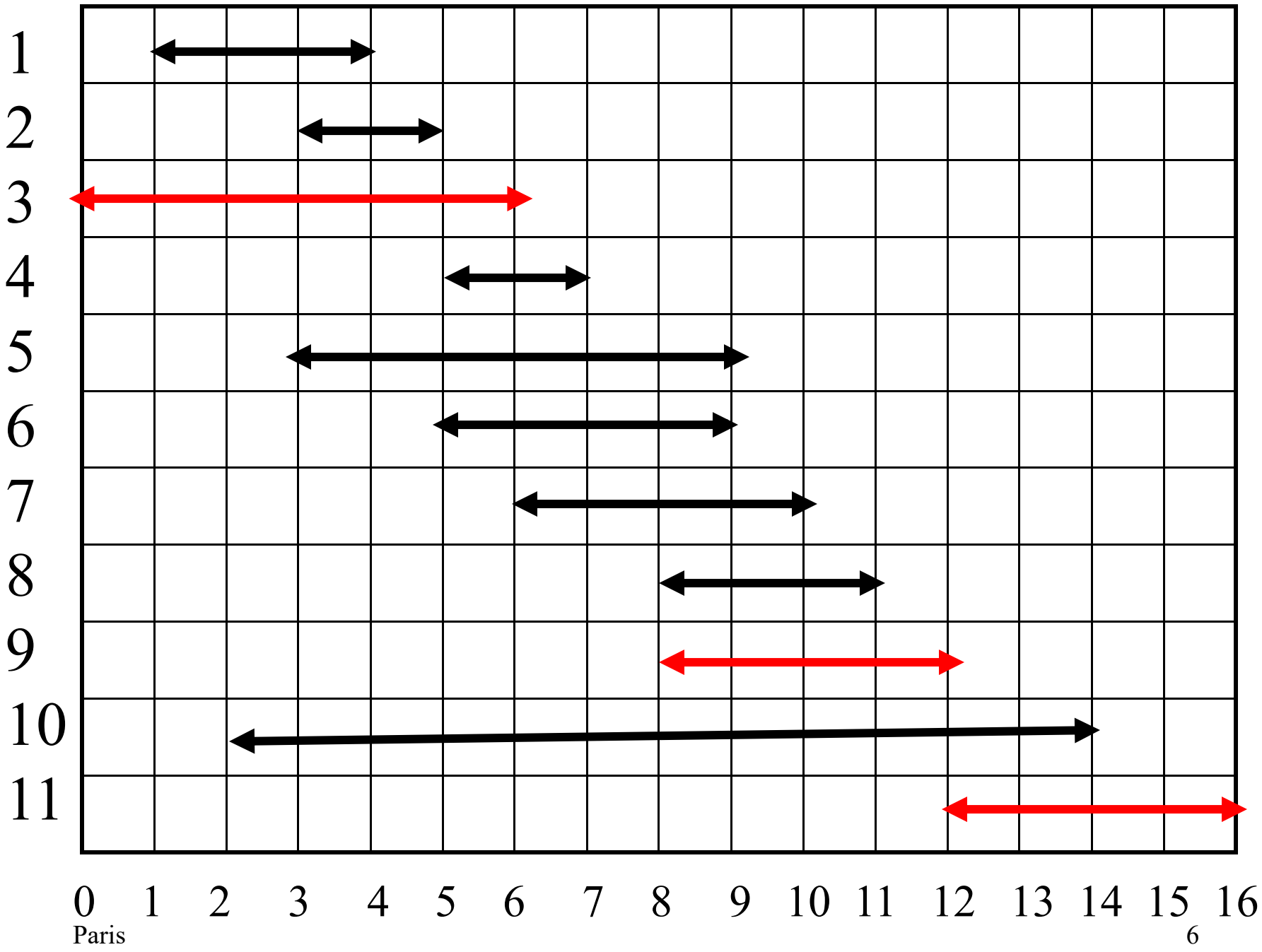
- What is the maximum number of activities that can be completed?
  - $\{a_3, a_9, a_{11}\}$ can be completed
  - But so can $\{a_1, a_4, a_8, a_{11}\}$ which is a larger set
  - But it is not unique, consider $\{a_2, a_4, a_9, a_{11}\}$

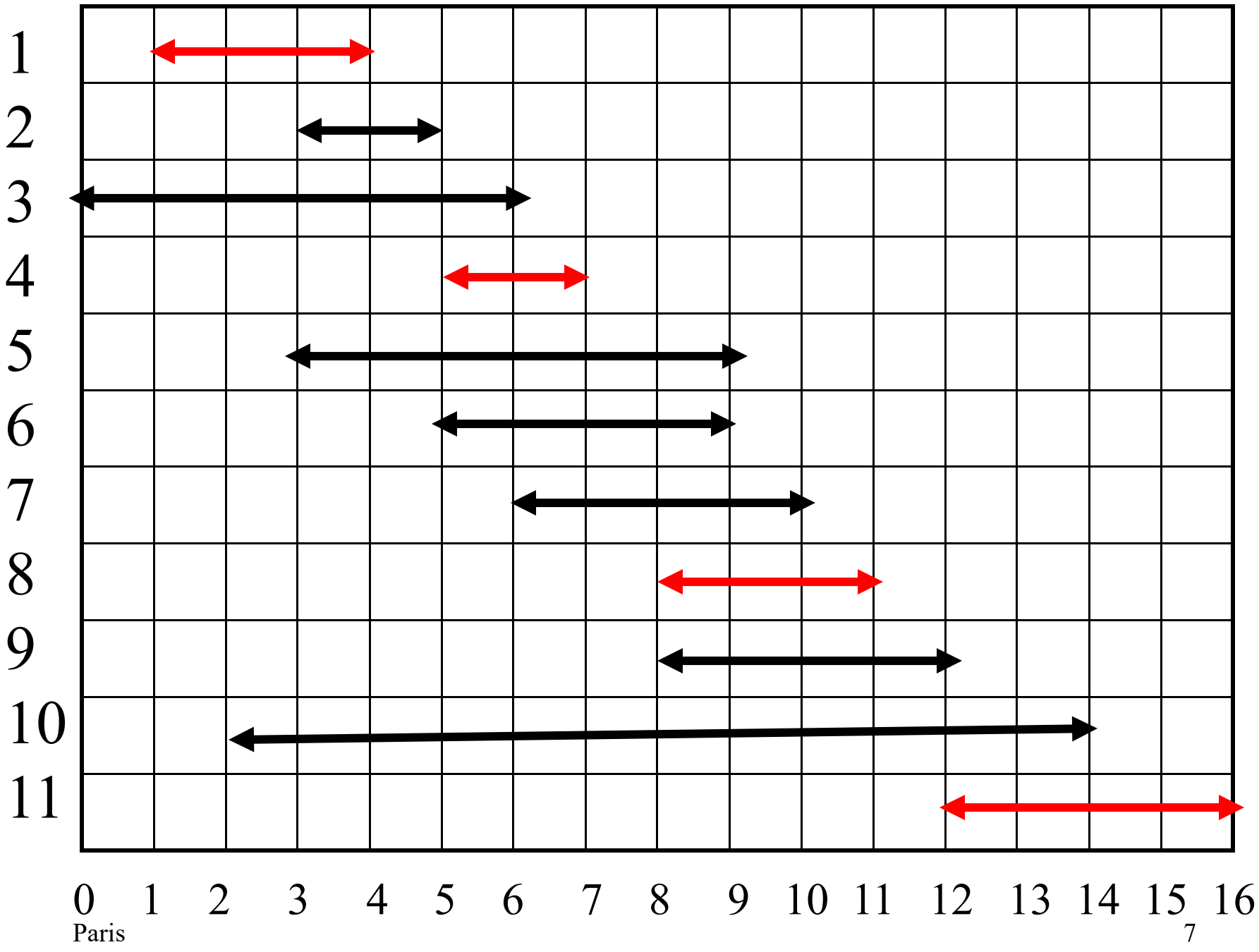
Interval Representation

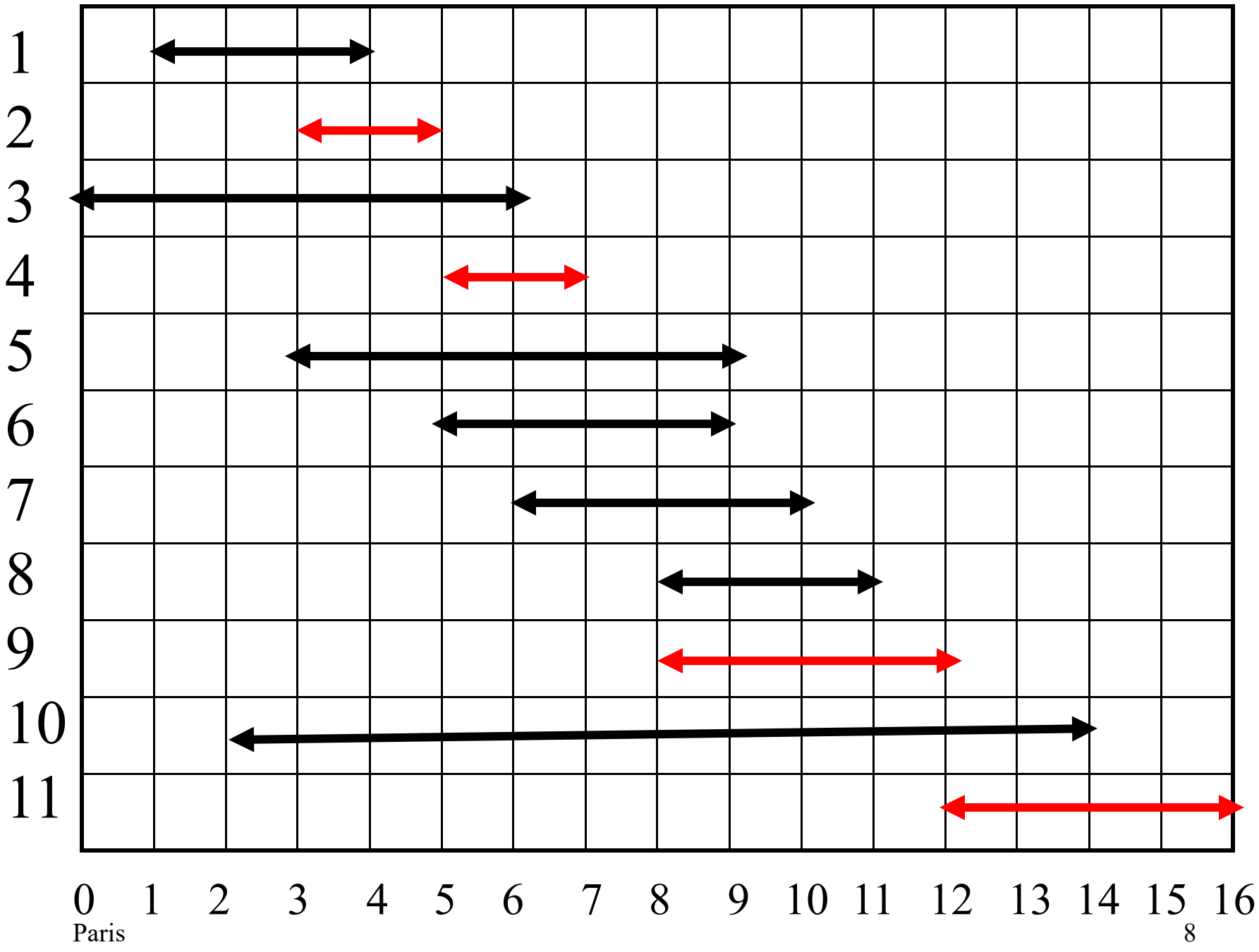
Activity No (i)	1	2	3	4	5	6	7	8	9	10	11
Start time (s_i)	1	3	0	5	3	5	6	8	8	2	12
Finish time (f_i)	4	5	6	7	9	9	10	11	12	14	16




1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

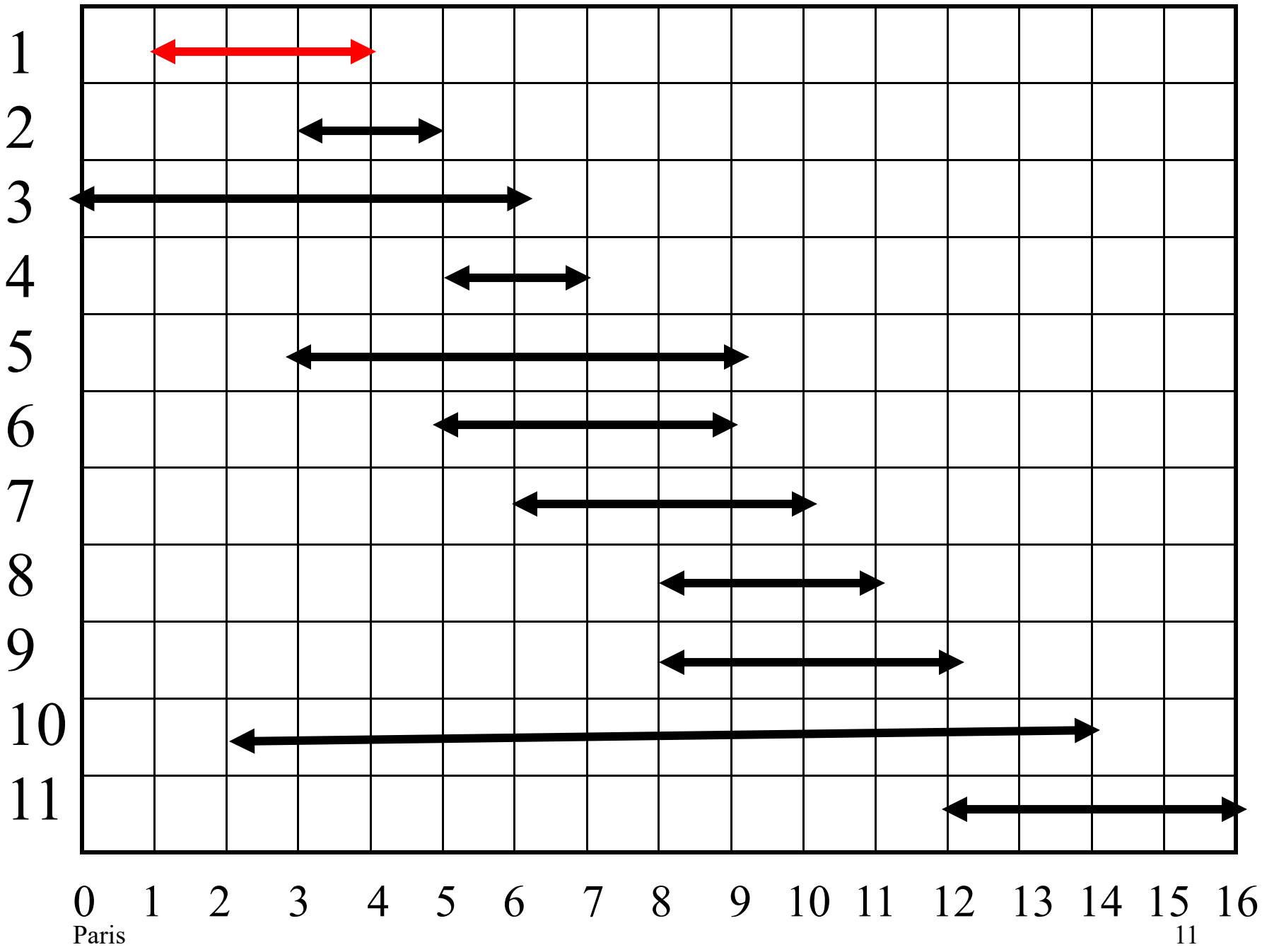

1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

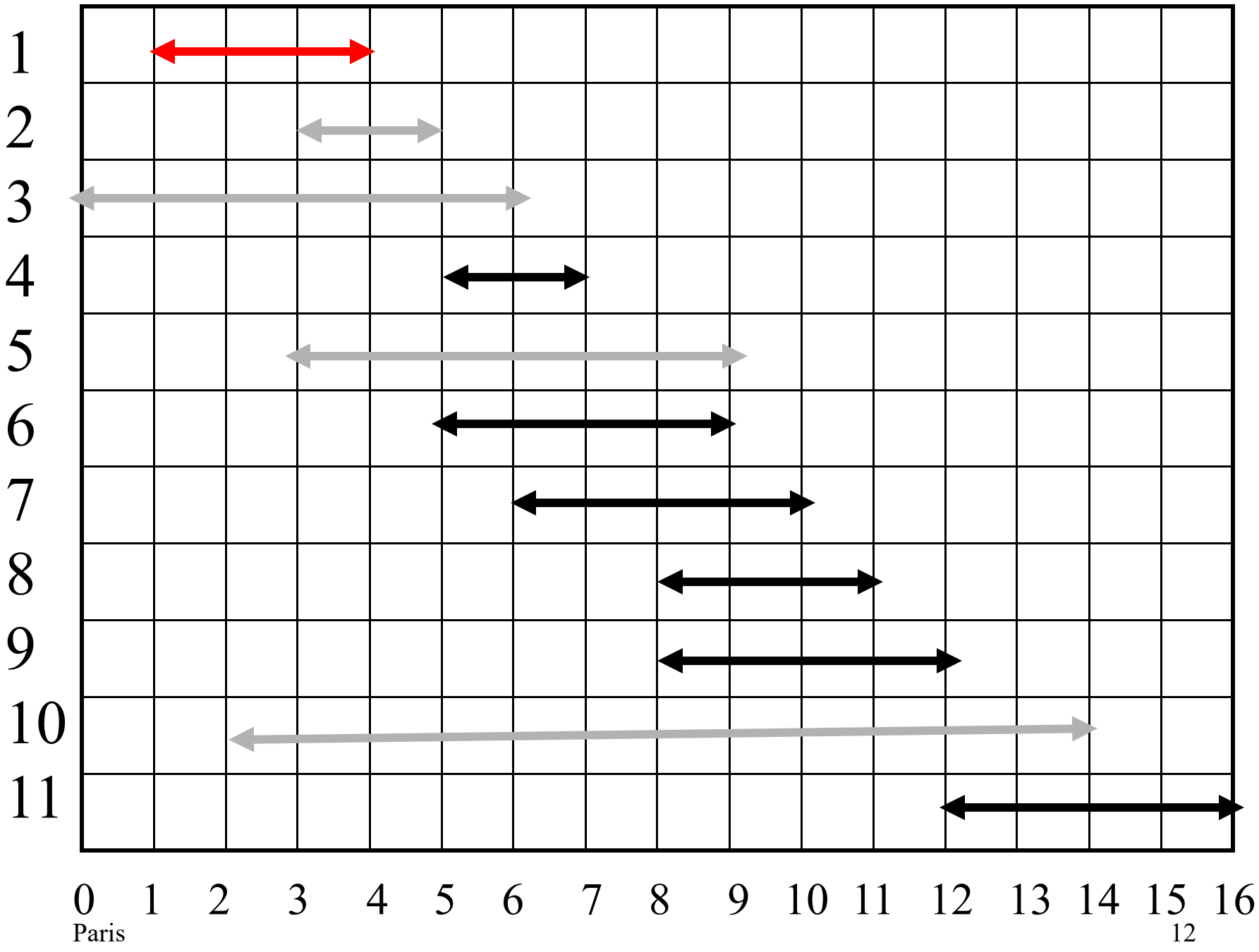
Greedy-choice property: Early Finish

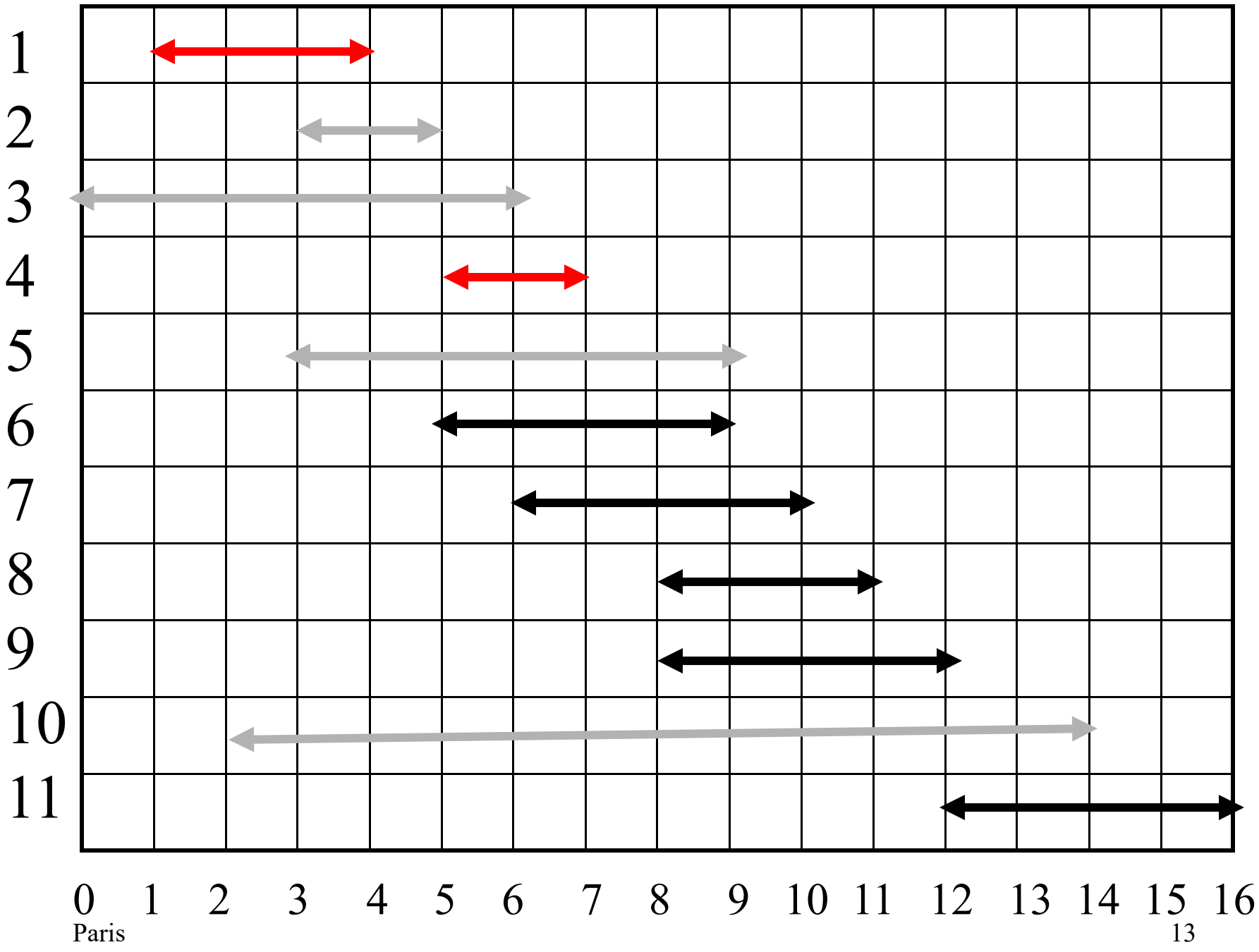
- Select the activity with the earliest finish
- Eliminate the activities that could not be scheduled
- Repeat

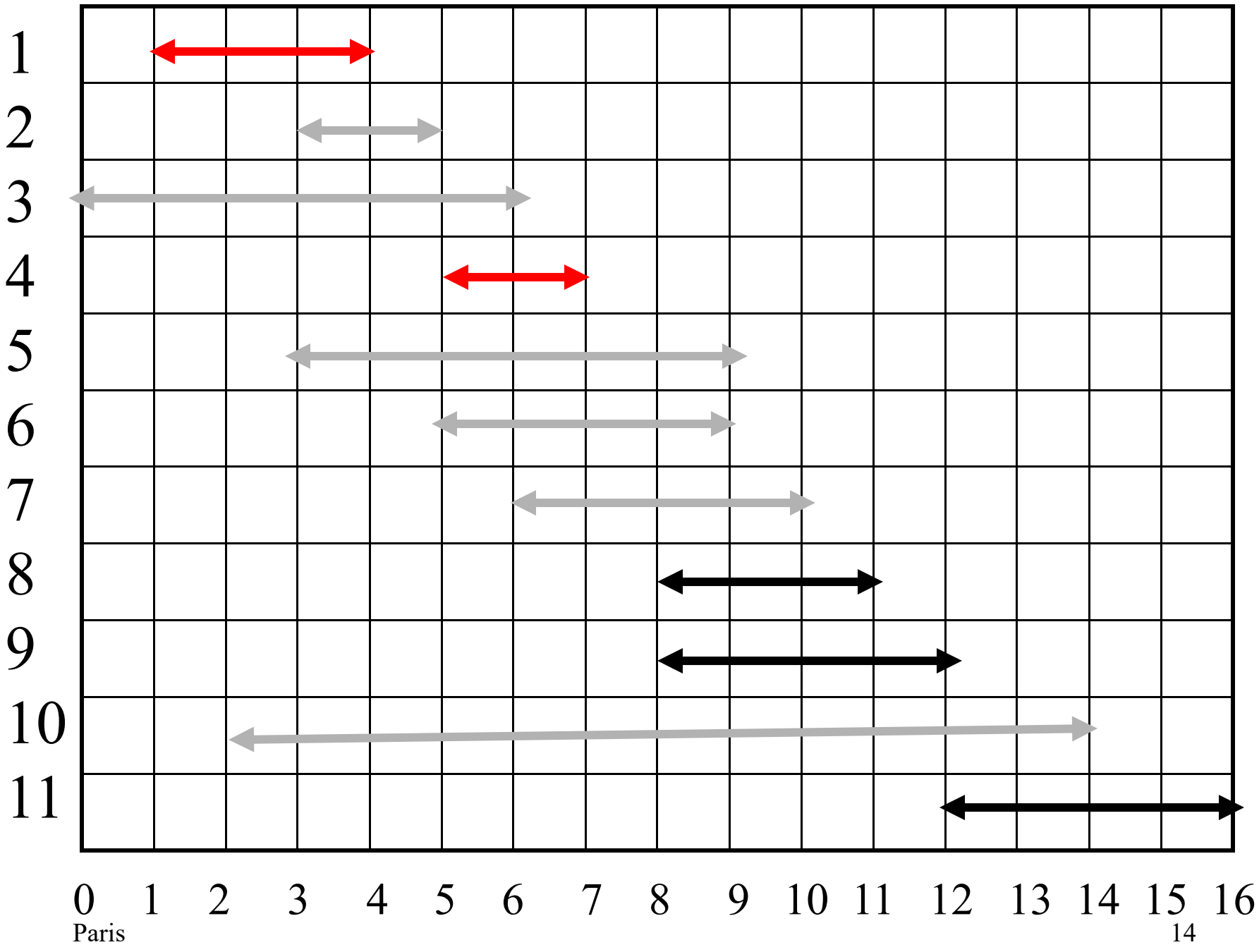
Why Greedy?

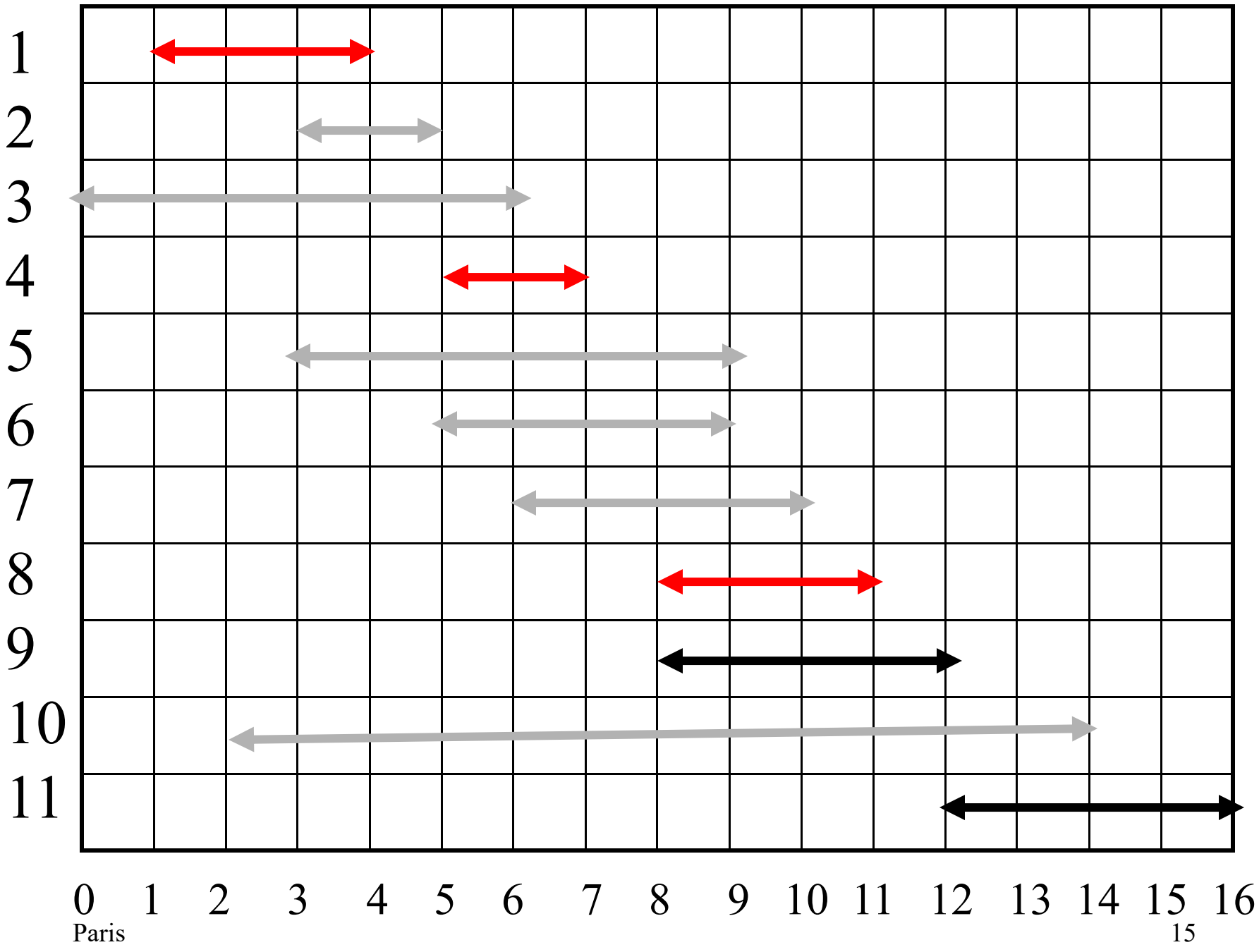
- Greedy in the sense that it leaves as much opportunity as possible for the remaining activities to be scheduled
- The greedy choice is the one that maximizes the amount of unscheduled time remaining

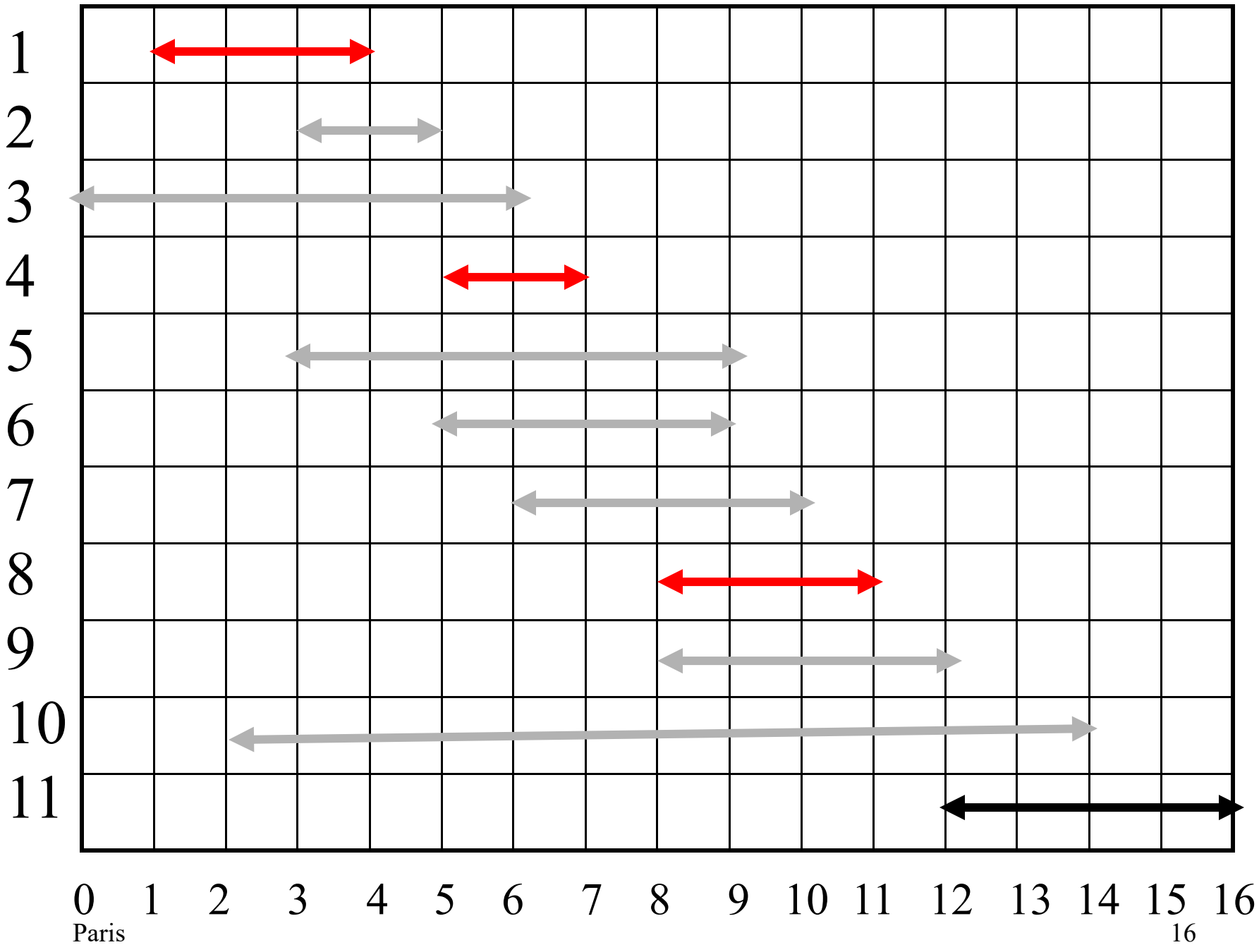

1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

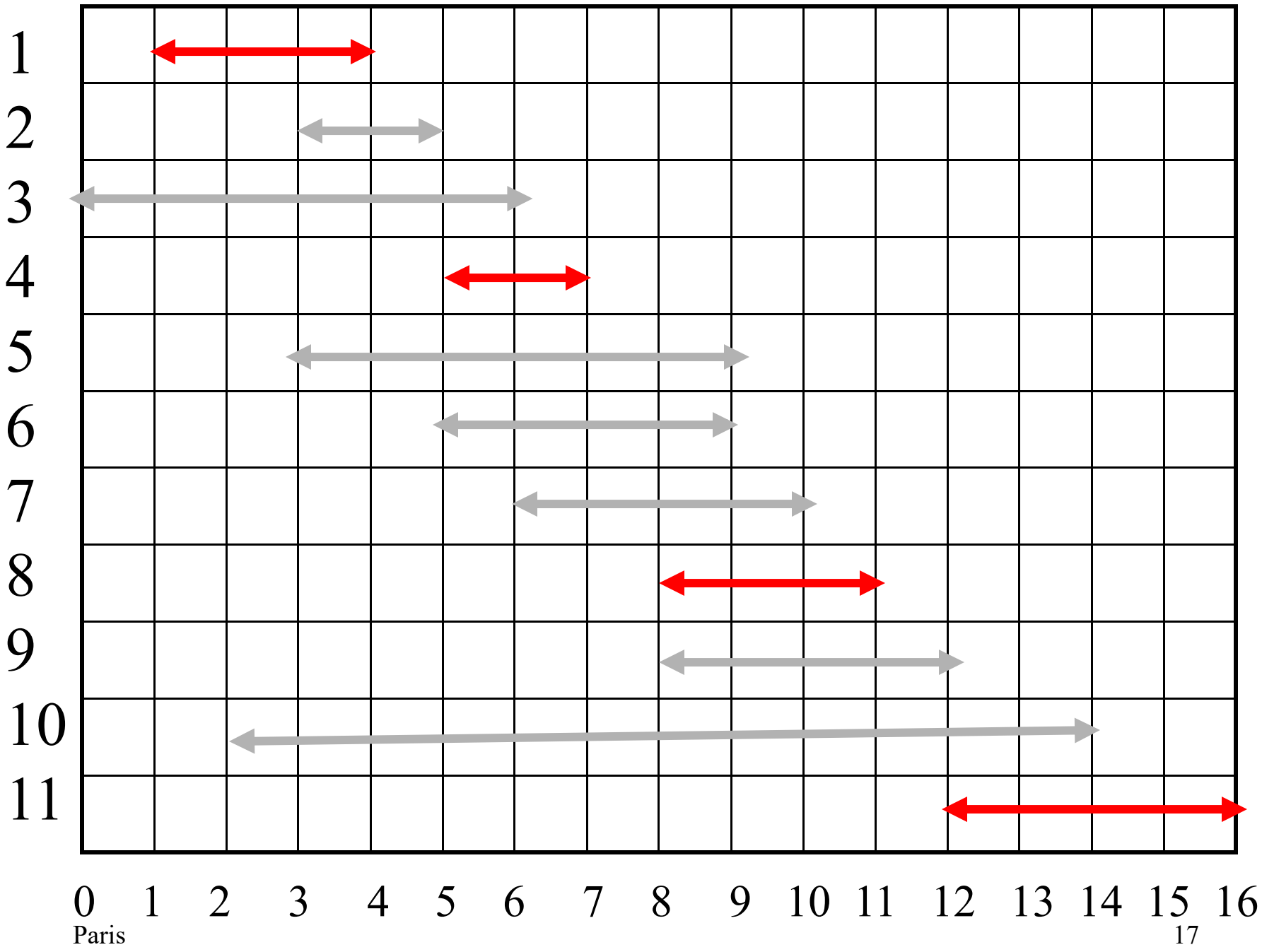

1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


1
2
3
4
5
6
7
8
9
10
11
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Algorithm : Recursive (Page-419)

//Assuming activities are sorted by finish time

RECURSIVE-ACTIVITY-SELECTOR$(s, f, 0, n)$.

RECURSIVE-ACTIVITY-SELECTOR(s, f, k, n)

```
1  m = k + 1
2  while m ≤ n and s[m] < f[k]        // find the first activity in S_k to fi
3        m = m + 1
4  if m ≤ n
5        return {a_m} ∪ RECURSIVE-ACTIVITY-SELECTOR(s, f, m, n)
6  else return ∅
```

Algorithm : Non Recursive (Page-421)

//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
```

Thank You

Stay Safe