

Algorithms

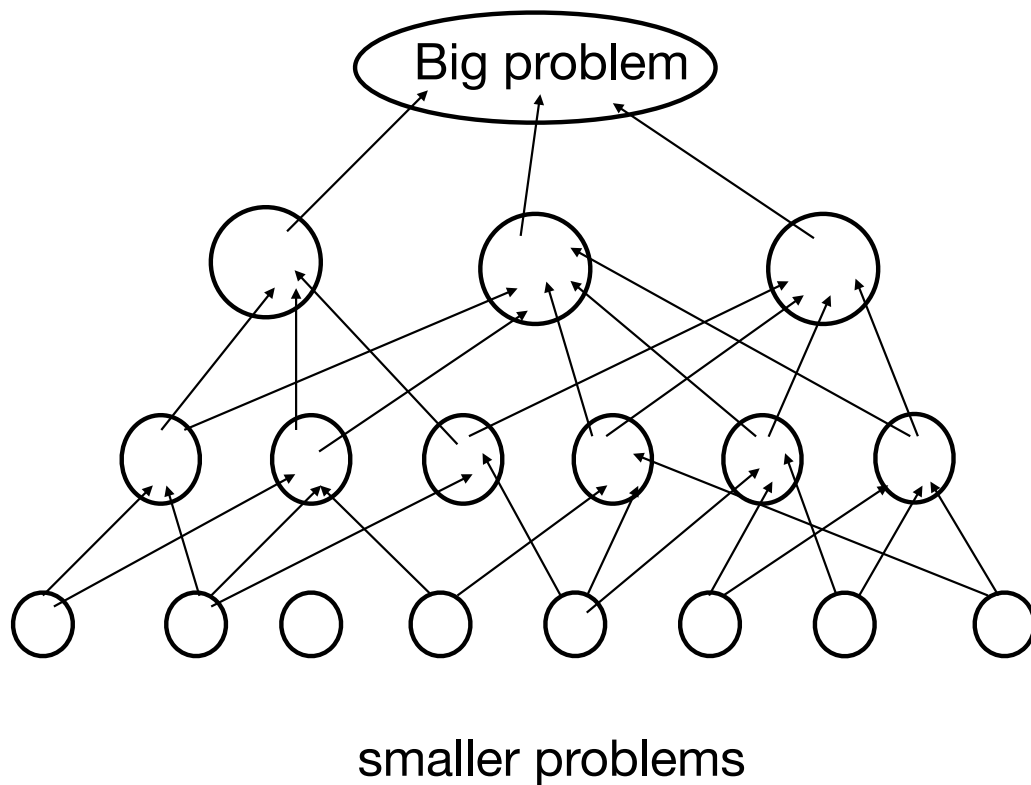
Lecture 3: Greedy Algorithm

Anxiao (Andrew) Jiang

CH 16. Greedy Algorithms

Comparison between “Dynamic Programming” and “Greedy Algorithm”:

Dynamic Programming



Greedy Algorithm

Make a sequence of greedy (locally optimal) choices to reach a solution.

16.1 Activity Selection Problem

Input: n activities a_1, a_2, \dots, a_n

For $i = 1, 2, \dots, n$, the activity a_i has **start time** s_i and **finish time** f_i

Output: A maximum subset of activities that are **compatible in time**.

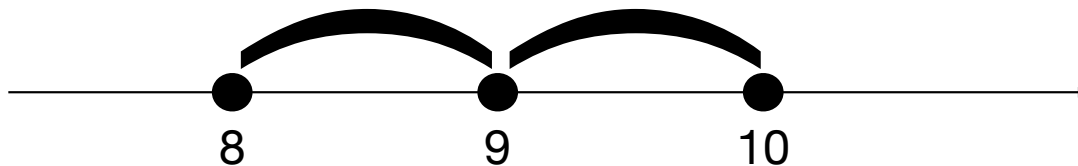
16.1 Activity Selection Problem

Input: n activities a_1, a_2, \dots, a_n

For $i = 1, 2, \dots, n$, the activity a_i has **start time** s_i and **finish time** f_i

Output: A maximum subset of activities that are **compatible in time**.

Are these two activities compatible in time? Yes.



To be mathematically rigorous, let's say the time interval for activity a_i is $[s_i, f_i)$

16.1 Activity Selection Problem

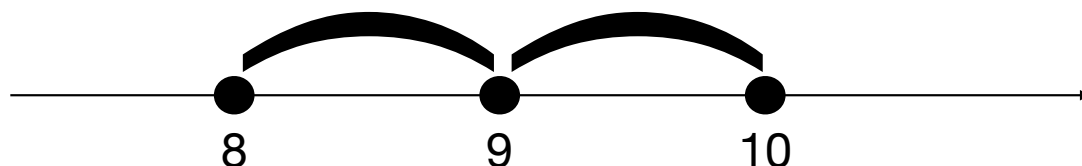
Input: n activities a_1, a_2, \dots, a_n

For $i = 1, 2, \dots, n$, the activity a_i has **start time** s_i and **finish time** f_i

Let's assume the n activities are sorted by their finish time: $f_1 \leq f_2 \leq \dots \leq f_n$

Output: A maximum subset of activities that are **compatible in time**.

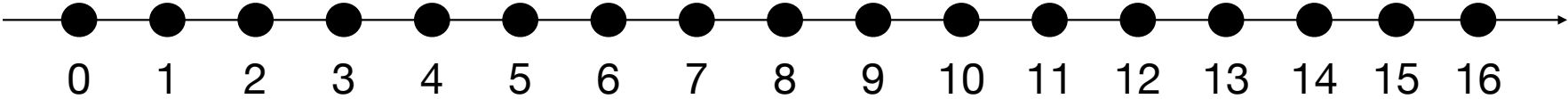
Are these two activities compatible in time? Yes.



To be mathematically rigorous, let's say the time interval for activity a_i is $[s_i, f_i)$

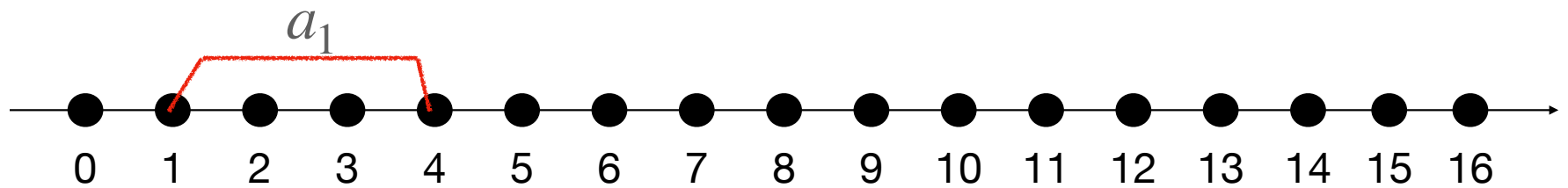
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



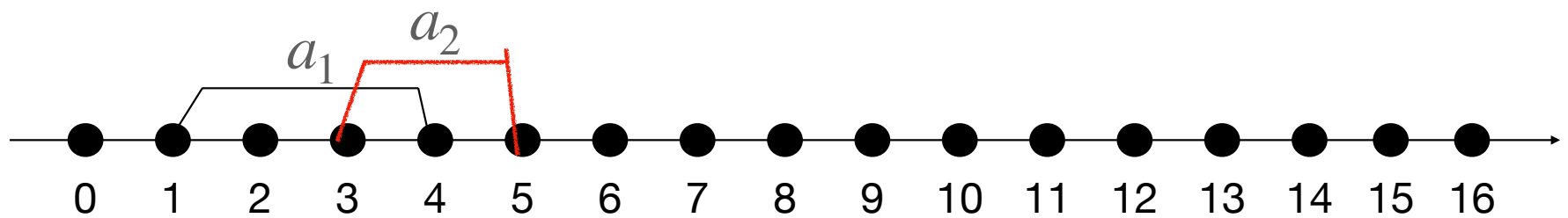
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



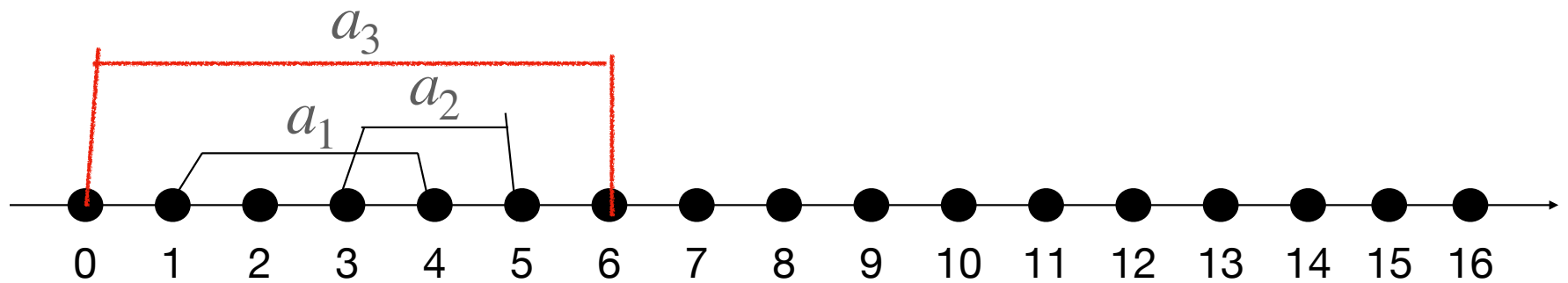
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



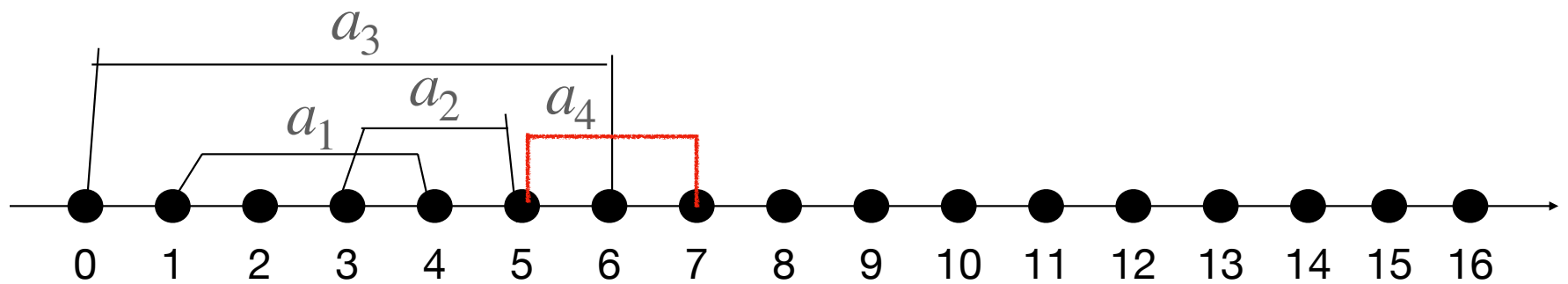
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



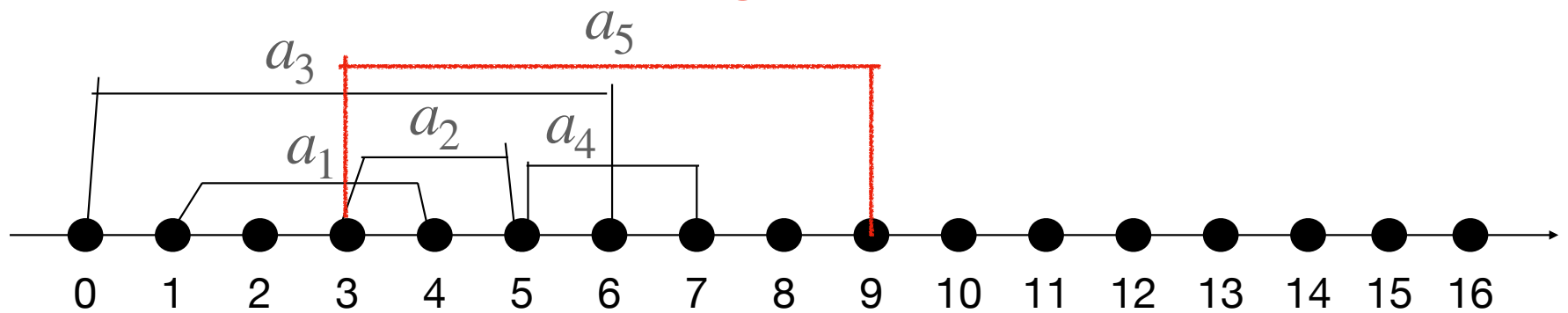
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



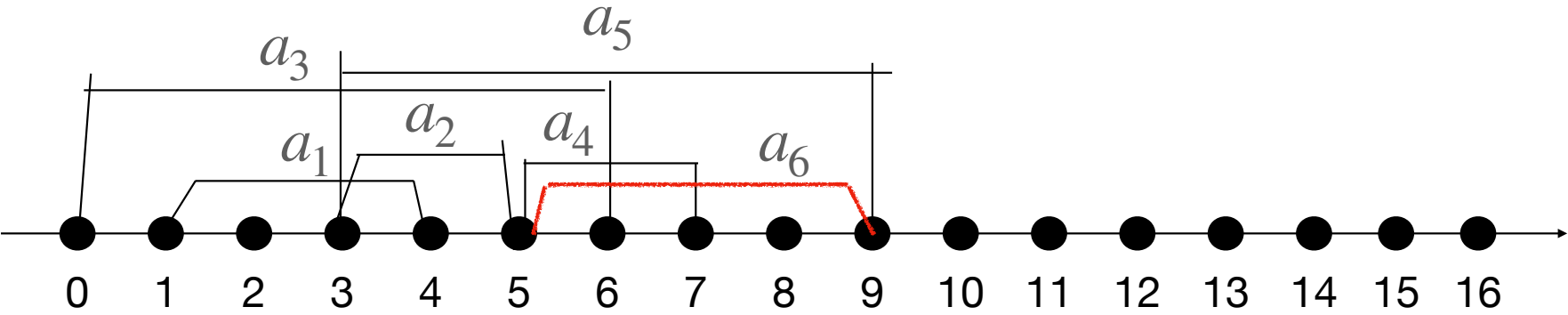
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



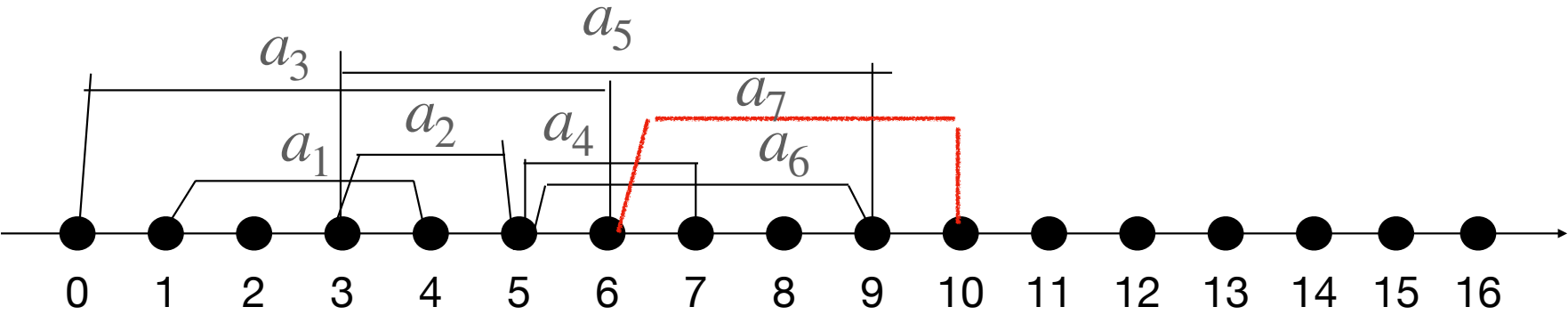
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



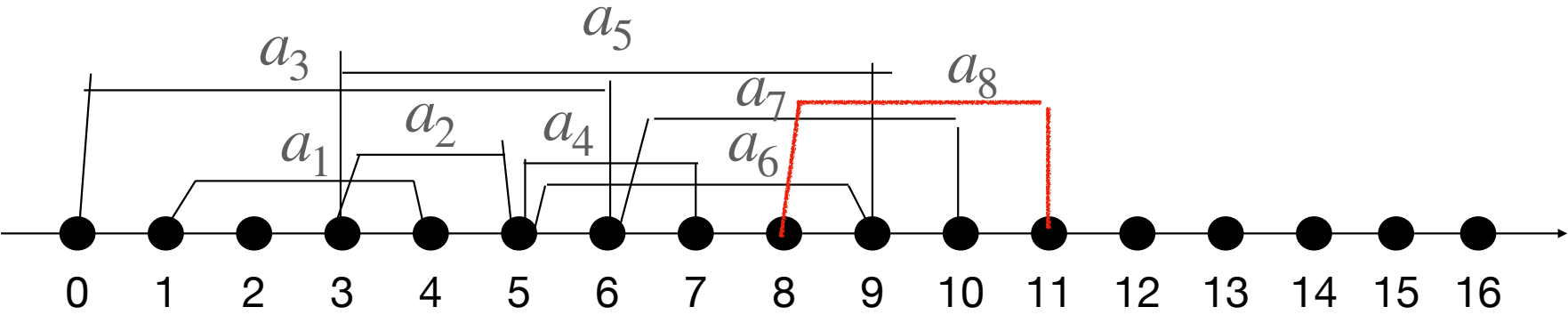
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



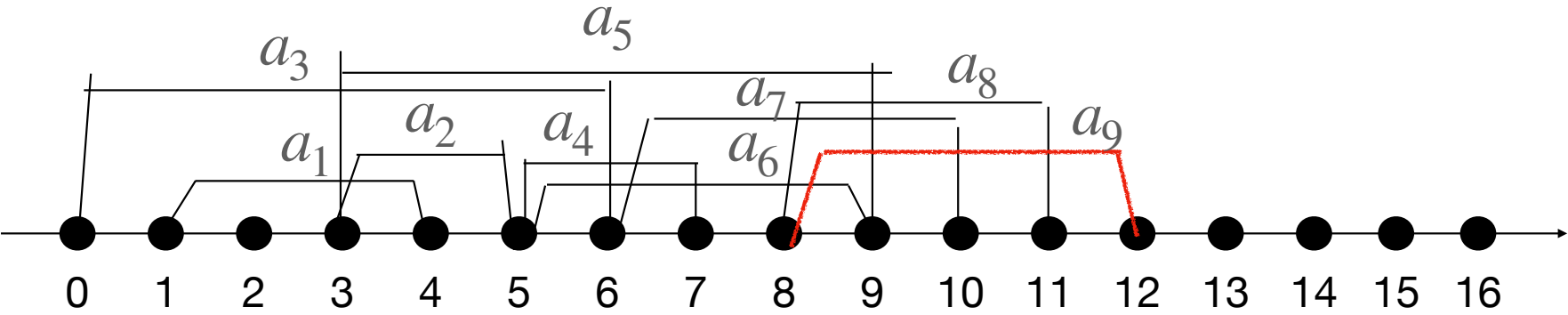
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



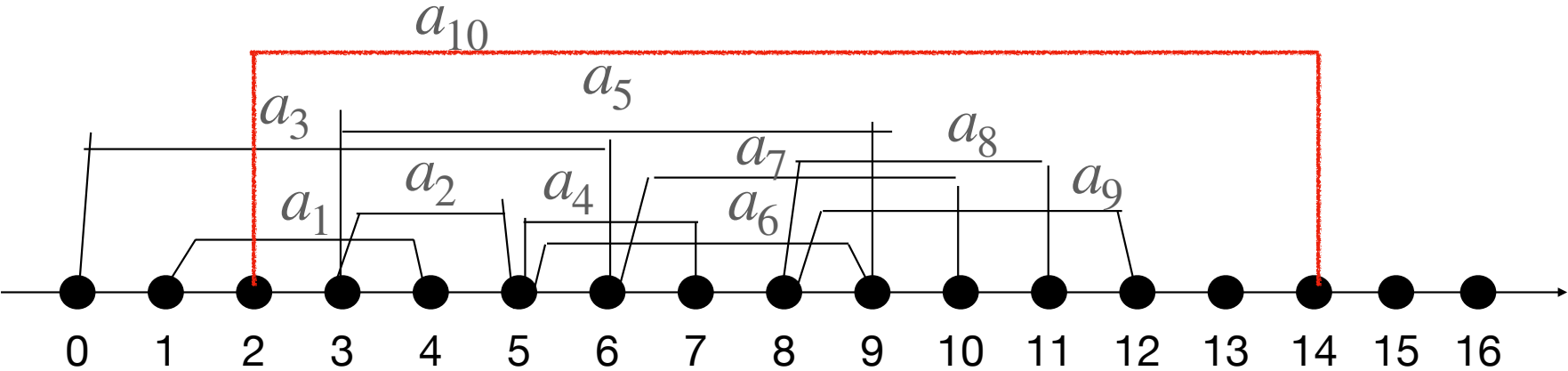
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



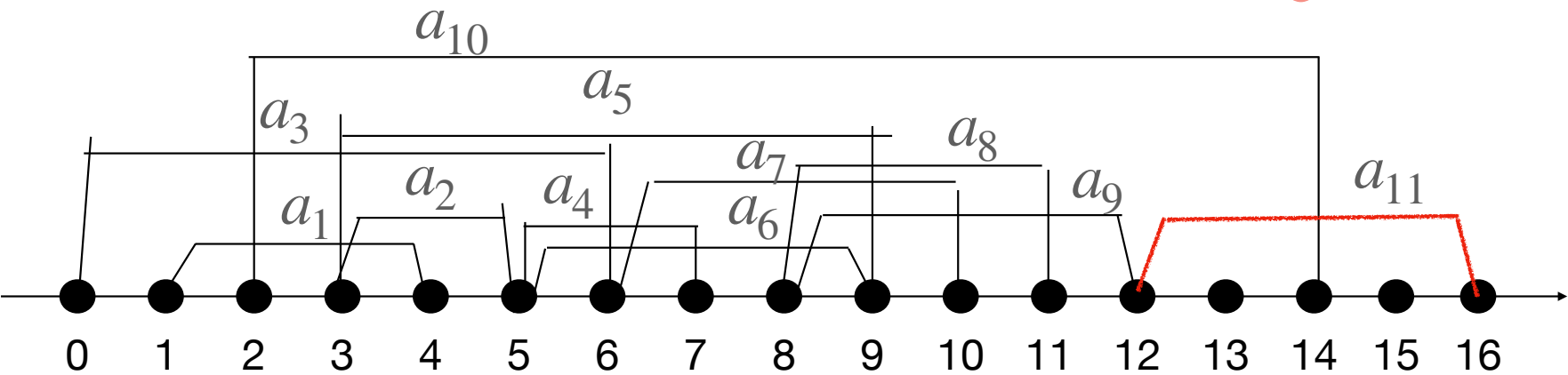
16.1 Activity Selection Problem

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

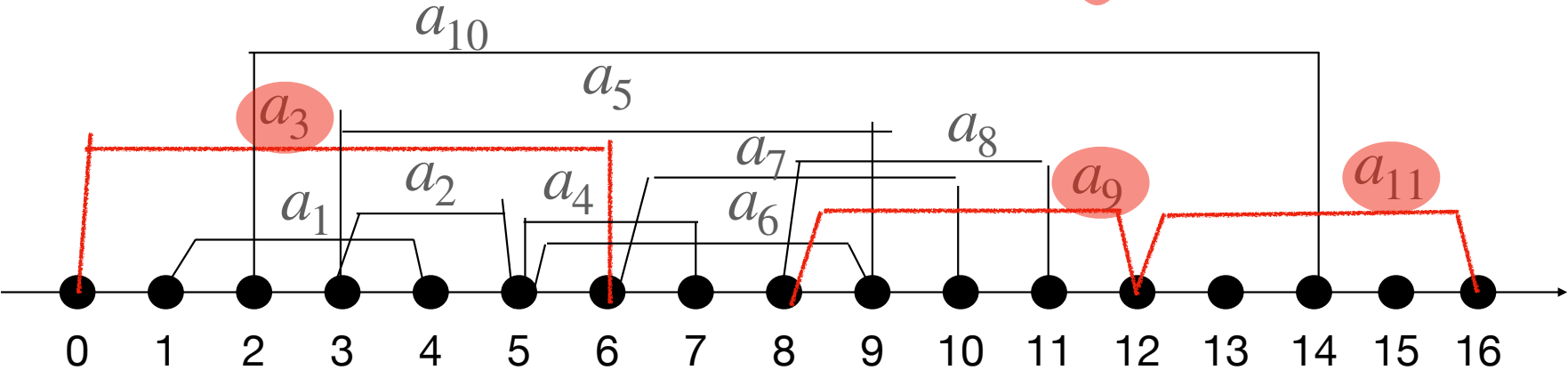
i	1	2	3	4	5	6	7	8	9	10	11
S_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16



16.1 Activity Selection Problem

Solution of
3 activities:
Not optimal

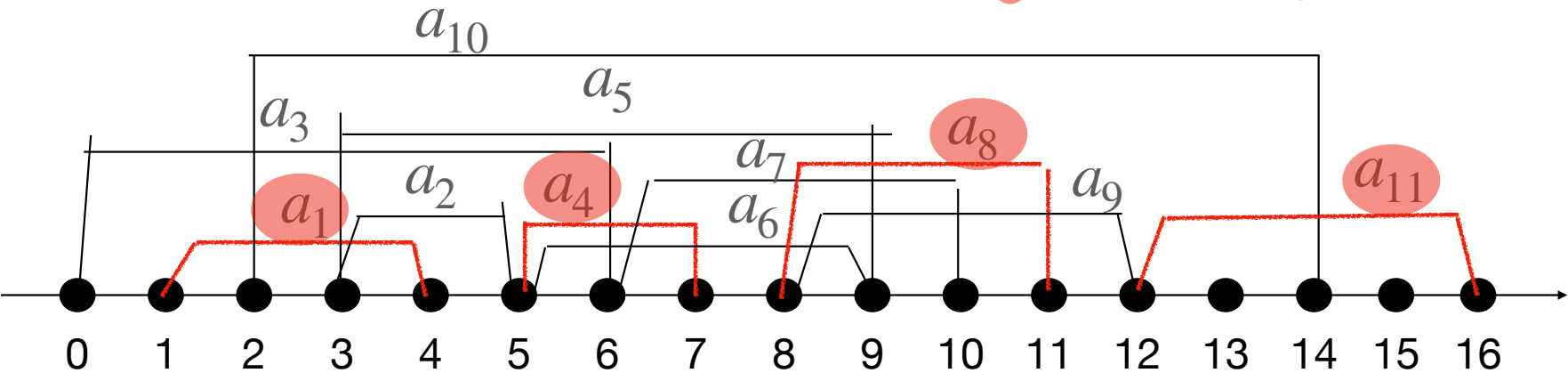
i	1	2	3	4	5	6	7	8	9	10	11
S_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16



16.1 Activity Selection Problem

A solution of
4 activities:
Optimal

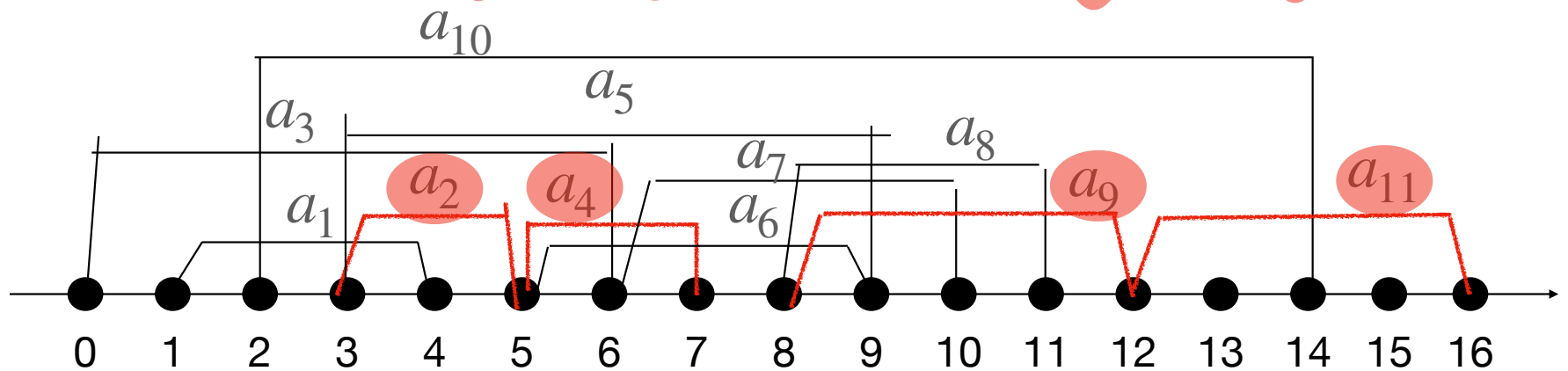
i	1	2	3	4	5	6	7	8	9	10	11
S_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16



16.1 Activity Selection Problem

Another solution of
 4 activities:
 Also optimal

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	9	9	10	11	12	14	16



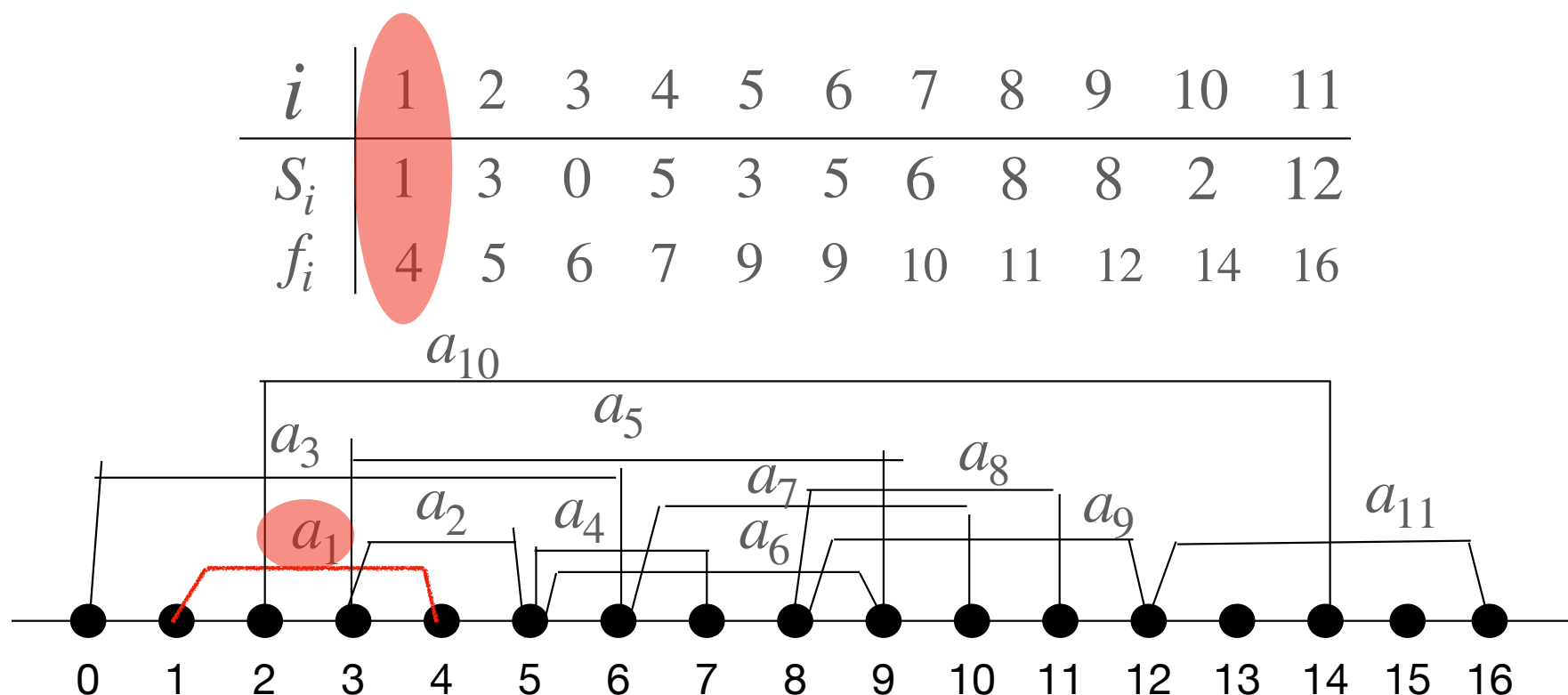
16.1 Activity Selection Problem

ANY ideas?

16.1 Activity Selection Problem

Idea of the greedy algorithm:

1) Pick the activity that **finishes first**.

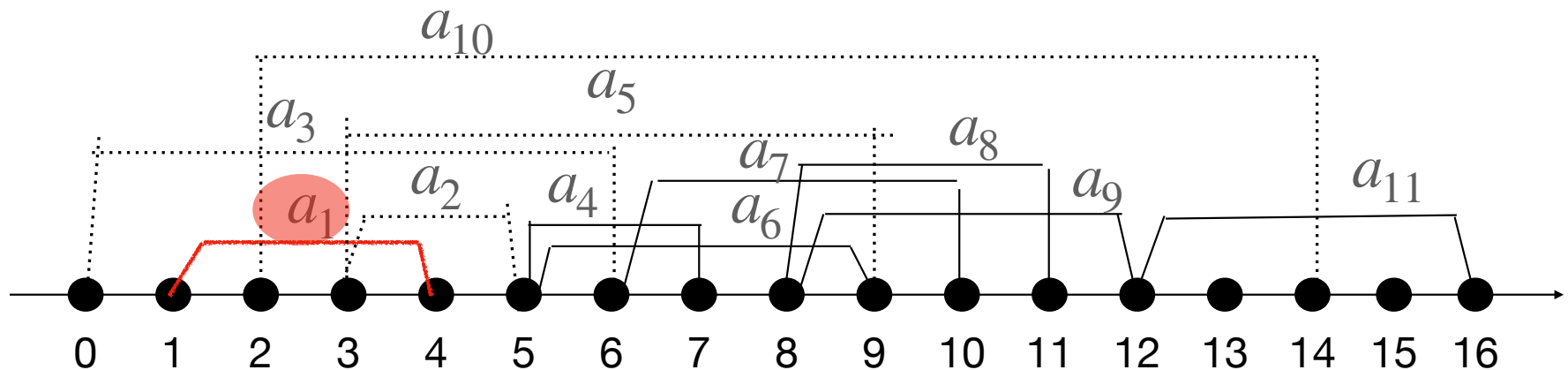


16.1 Activity Selection Problem

Idea of the greedy algorithm:

- 1) Pick the activity that **finishes first**.
- 2) Among the activities that are compatible with the chosen activity, pick as many compatible activities as possible.

Solid lines: activities compatible with a_1 . Select among them.



16.1 Activity Selection Problem

Idea of the greedy algorithm:

- 1) Pick the activity that finishes first. (which is a_1 , since activities are sorted by finish time.)
- 2) Among the activities that are compatible with the chosen activity, pick as many compatible activities as possible.

Input: n activities a_1, a_2, \dots, a_n

For $i = 1, 2, \dots, n$, the activity a_i has start time s_i and finish time f_i

Let's assume the n activities are sorted by their finish time: $f_1 \leq f_2 \leq \dots \leq f_n$

Output: A maximum subset of activities that are compatible in time.

16.1 Activity Selection Problem

Idea of the greedy algorithm:

- 1) Pick the activity that finishes first. (which is a_1 , since activities are sorted by finish time.)
 - 2) Among the activities that are compatible with the chosen activity, pick as many compatible activities as possible.
-

Lemma: There exists an optimal solution that contains a_1 .

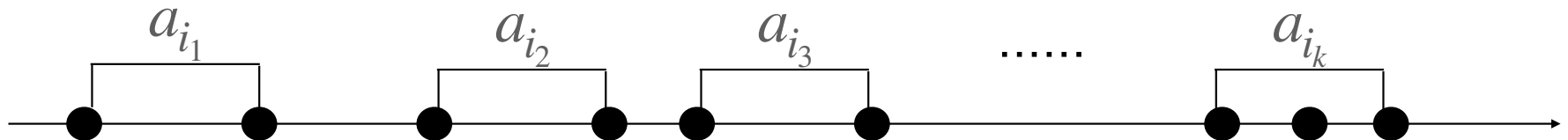
16.1 Activity Selection Problem

Idea of the greedy algorithm:

- 1) Pick the activity that **finishes first**. (which is a_1 , **since activities are sorted by finish time.**)
 - 2) Among the activities that are compatible with the chosen activity, pick as many compatible activities as possible.
-

Lemma: There exists an optimal solution that contains a_1 .

Proof: Consider an optimal solution



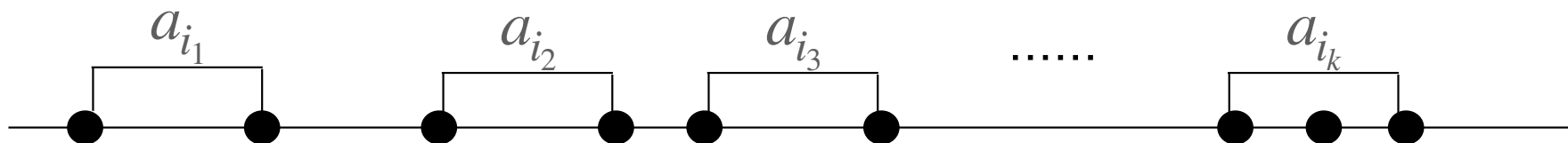
16.1 Activity Selection Problem

Idea of the greedy algorithm:

- 1) Pick the activity that **finishes first**. (which is a_1 , **since activities are sorted by finish time.**)
 - 2) Among the activities that are compatible with the chosen activity, pick as many compatible activities as possible.
-

Lemma: There exists an optimal solution that contains a_1 .

Proof: Consider an optimal solution



Where is a_1 ?

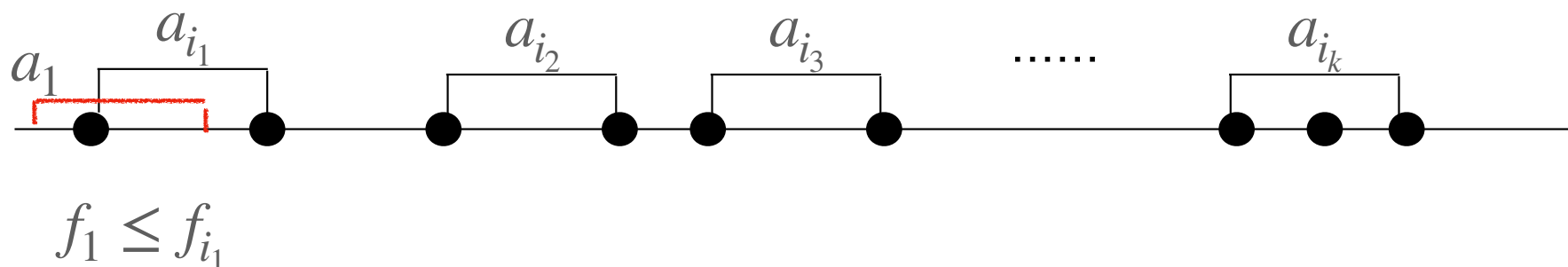
16.1 Activity Selection Problem

Idea of the greedy algorithm:

- 1) Pick the activity that **finishes first**. (which is a_1 , **since activities are sorted by finish time.**)
 - 2) Among the activities that are compatible with the chosen activity, pick as many compatible activities as possible.
-

Lemma: There exists an optimal solution that contains a_1 .

Proof: Consider an optimal solution



Where is a_1 ?

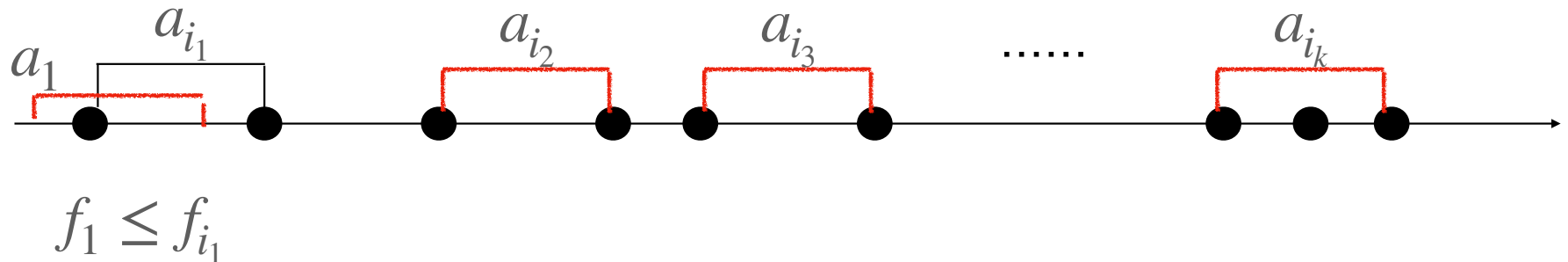
16.1 Activity Selection Problem

Idea of the greedy algorithm:

- 1) Pick the activity that **finishes first**. (which is a_1 , **since activities are sorted by finish time.**)
 - 2) Among the activities that are compatible with the chosen activity, pick as many compatible activities as possible.
-

Lemma: There exists an optimal solution that contains a_1 .

Proof: Consider an optimal solution



By replacing a_{i_1} with a_1 ,
we get an optimal solution that contains a_1 .

Every time we make a greedy choice, we have narrowed our solution space.

As long as the solution space still contains an optimal solution, it is all right.

From now on, we focus only on optimal solutions that are consistent with the greedy choices we have made.

16.1 Activity Selection Problem

Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. *Return* A

16.1 Activity Selection Problem

Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$ Here k is the index of the last selected (right most) activity.
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A

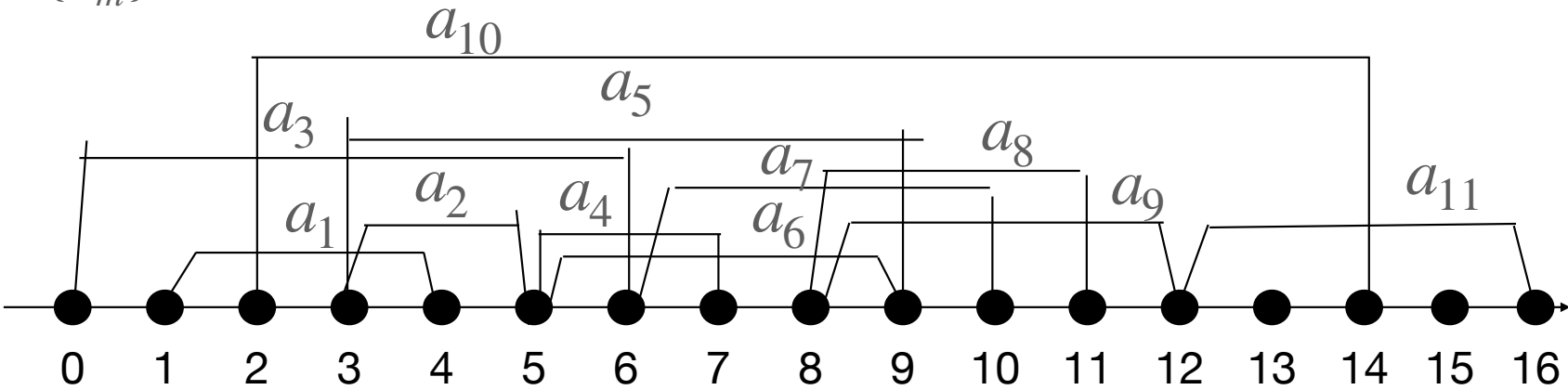
16.1 Activity Selection Problem

Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

- 1. $A = \{a_1\}$
- 2. $k = 1$
- 3. for $m = 2$ to n
- 4. if $s_m \geq f_k$
- 5. $A = A \cup \{a_m\}$
- 6. $k = m$
- 7. Return A

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

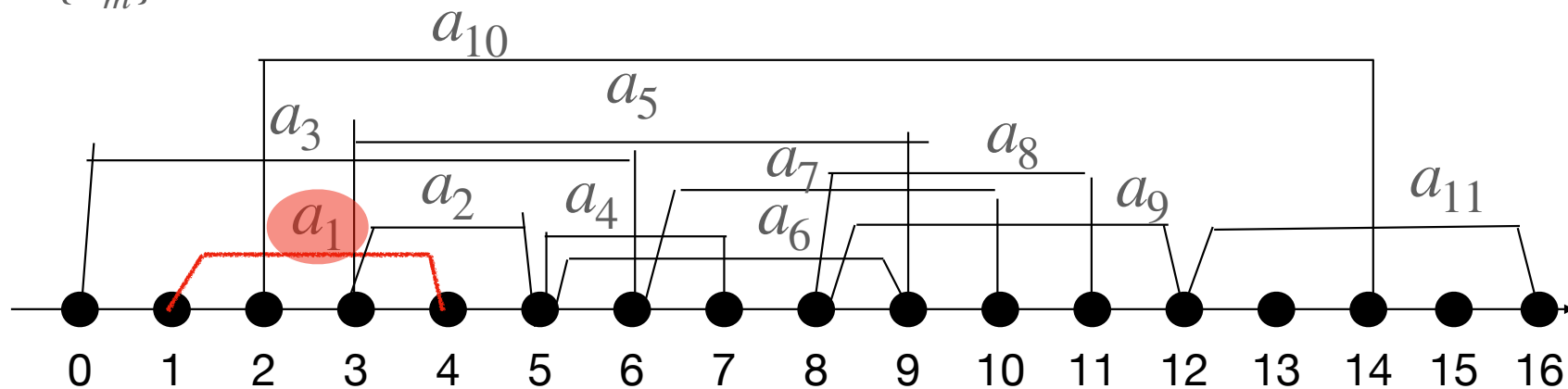
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A

$A = \{a_1\}$ $k = 1$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

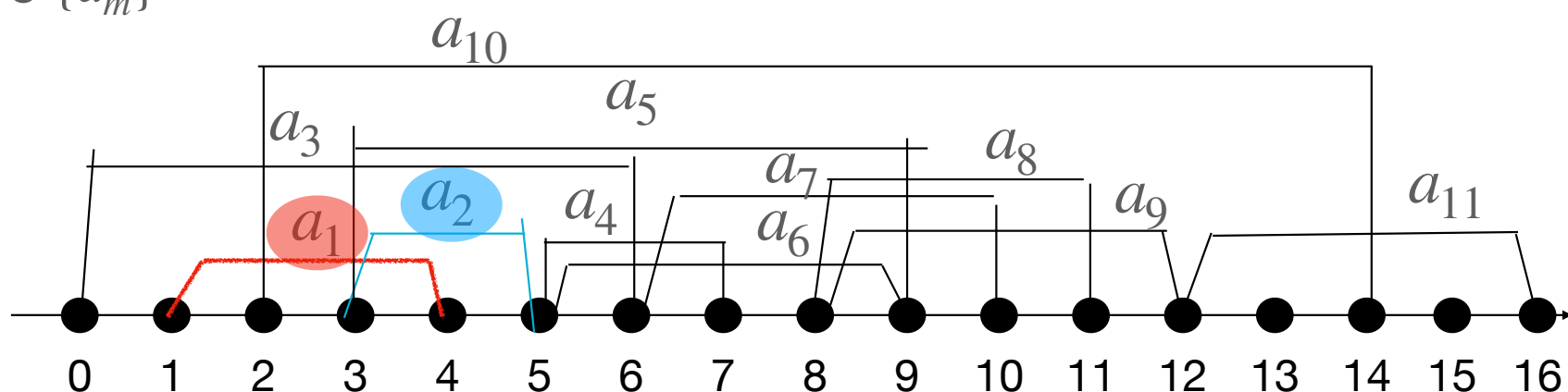
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A

$$A = \{a_1\} \quad k = 1$$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

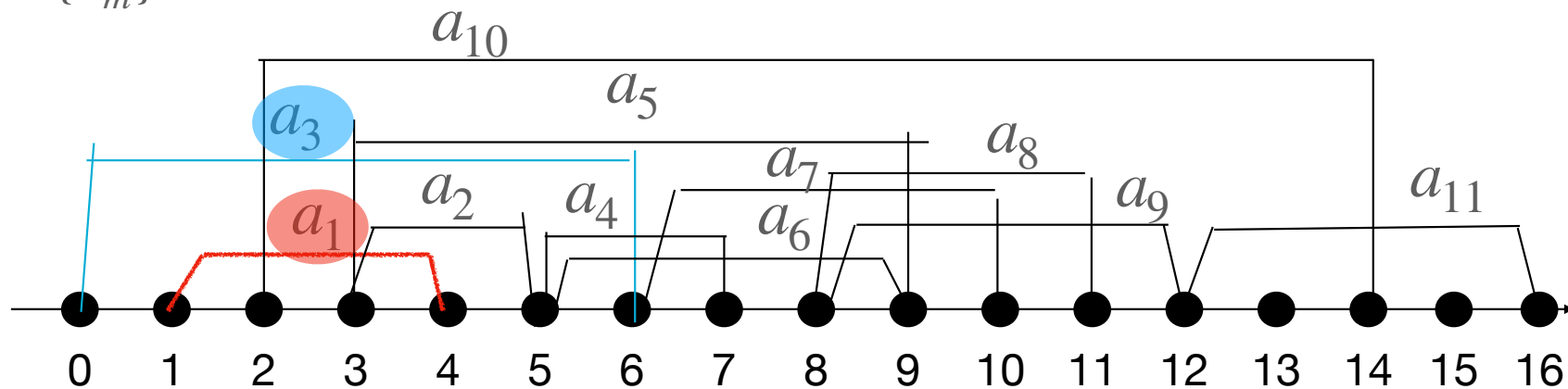
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A

$A = \{a_1\}$ $k = 1$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

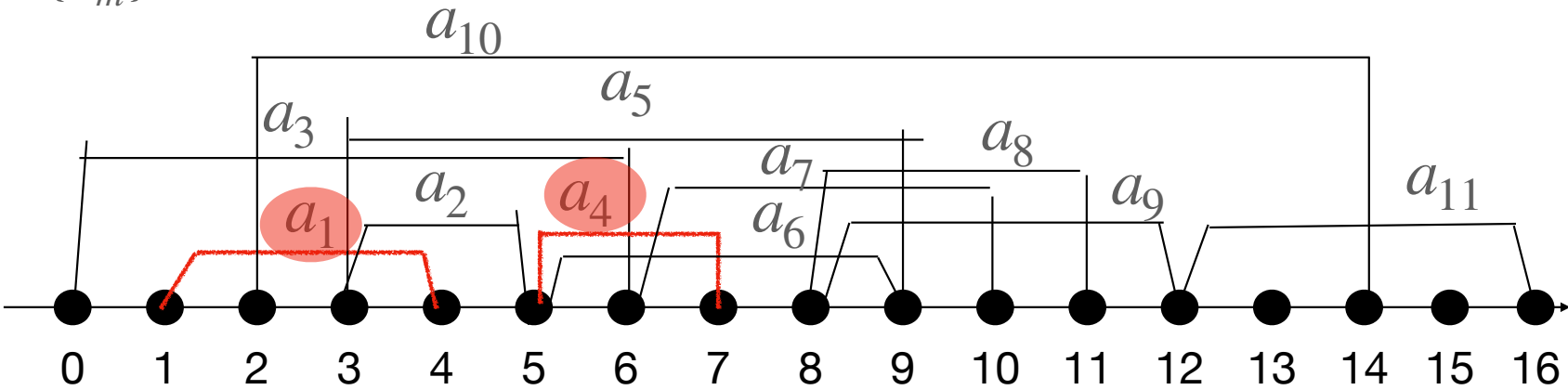
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

- 1. $A = \{a_1\}$
- 2. $k = 1$
- 3. for $m = 2$ to n
- 4. if $s_m \geq f_k$
- 5. $A = A \cup \{a_m\}$
- 6. $k = m$
- 7. Return A

$A = \{a_1, a_4\}$ $k = 4$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

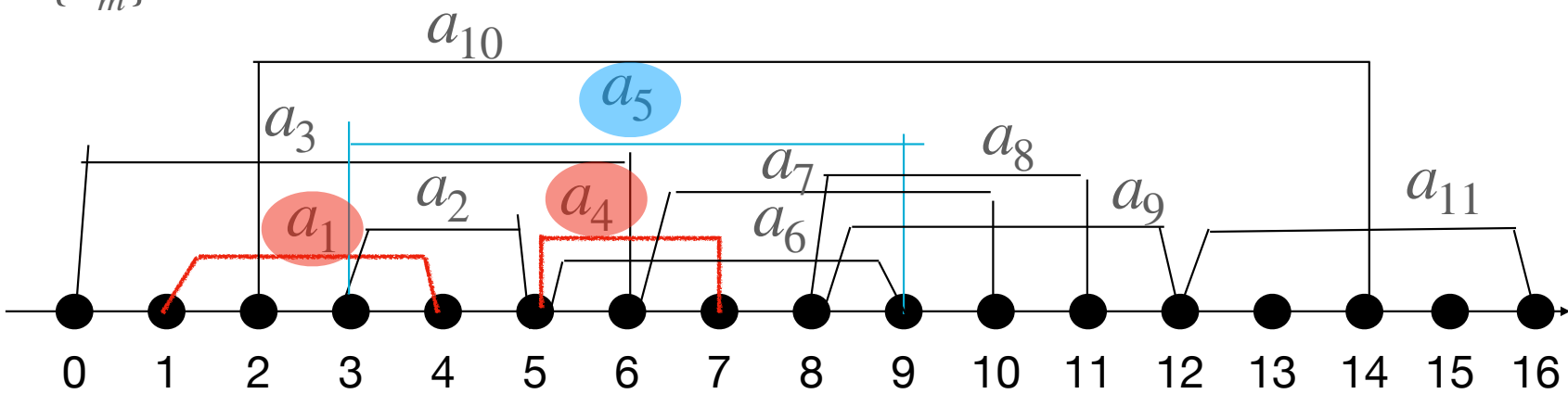
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

- 1. $A = \{a_1\}$
- 2. $k = 1$
- 3. for $m = 2$ to n
- 4. if $s_m \geq f_k$
- 5. $A = A \cup \{a_m\}$
- 6. $k = m$
- 7. Return A

$A = \{a_1, a_4\}$ $k = 4$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

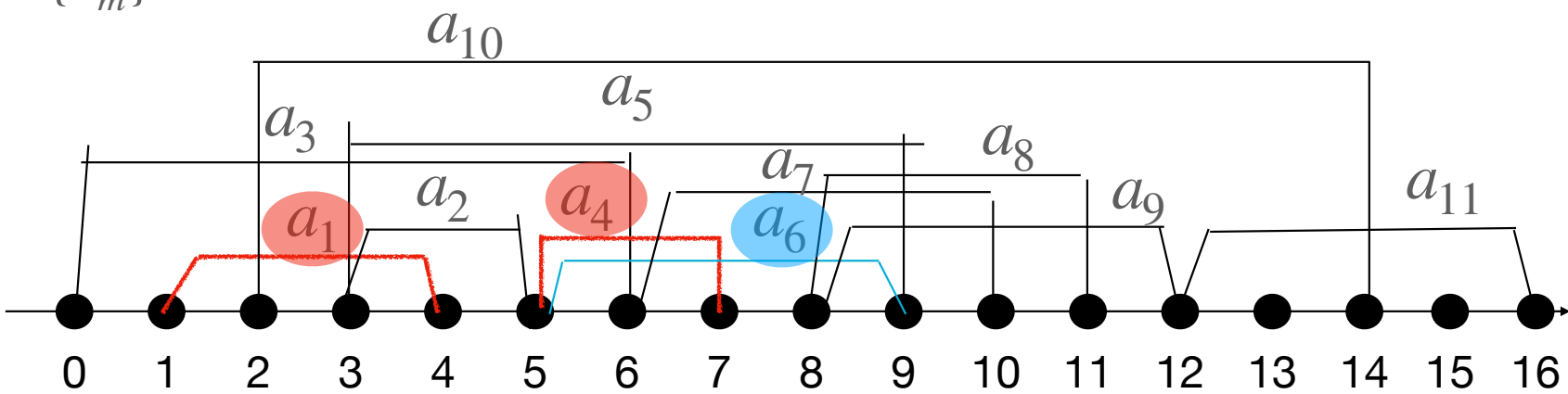
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

- 1. $A = \{a_1\}$
- 2. $k = 1$
- 3. for $m = 2$ to n
- 4. if $s_m \geq f_k$
- 5. $A = A \cup \{a_m\}$
- 6. $k = m$
- 7. Return A

$A = \{a_1, a_4\}$ $k = 4$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

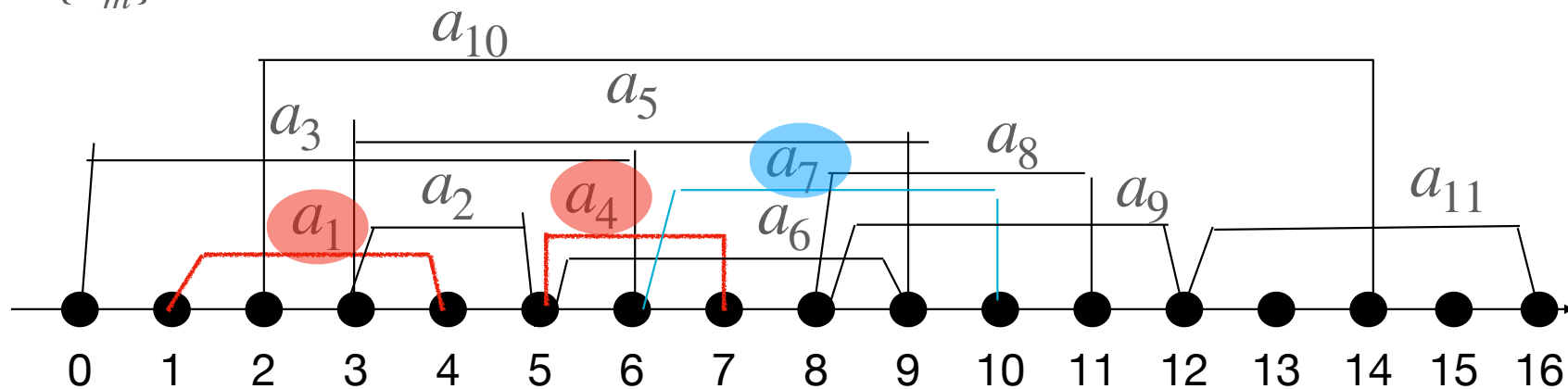
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A

$$A = \{a_1, a_4\} \quad k = 4$$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

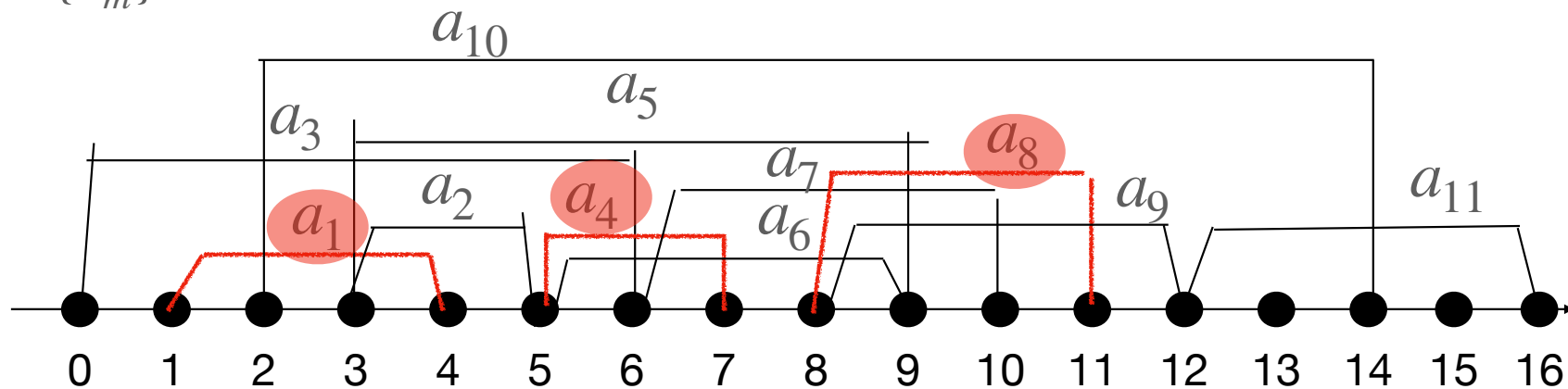
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A

$$A = \{a_1, a_4, a_8\} \quad k = 8$$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

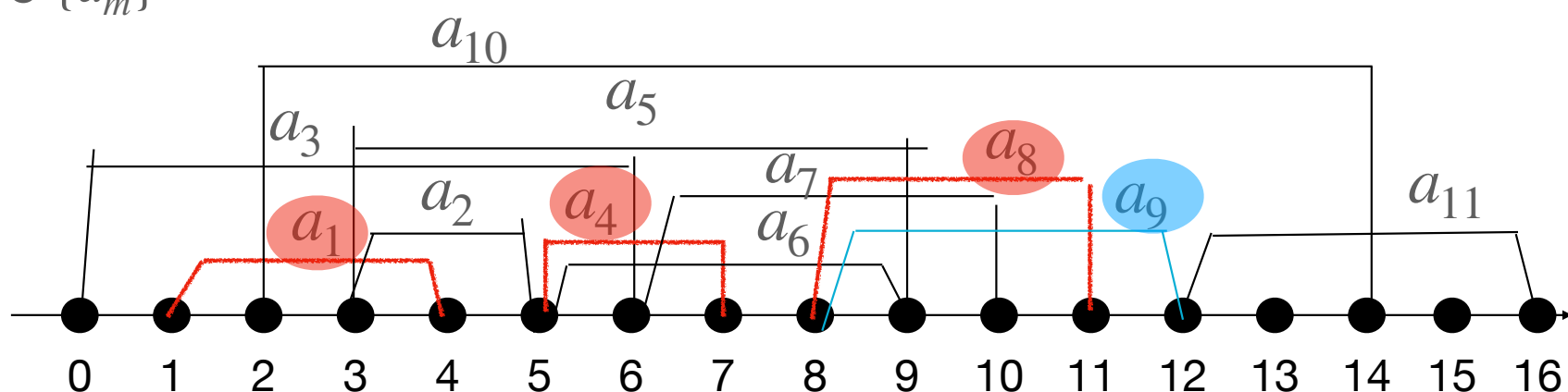
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A

$$A = \{a_1, a_4, a_8\} \quad k = 8$$

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	9	9	10	11	12	14	16



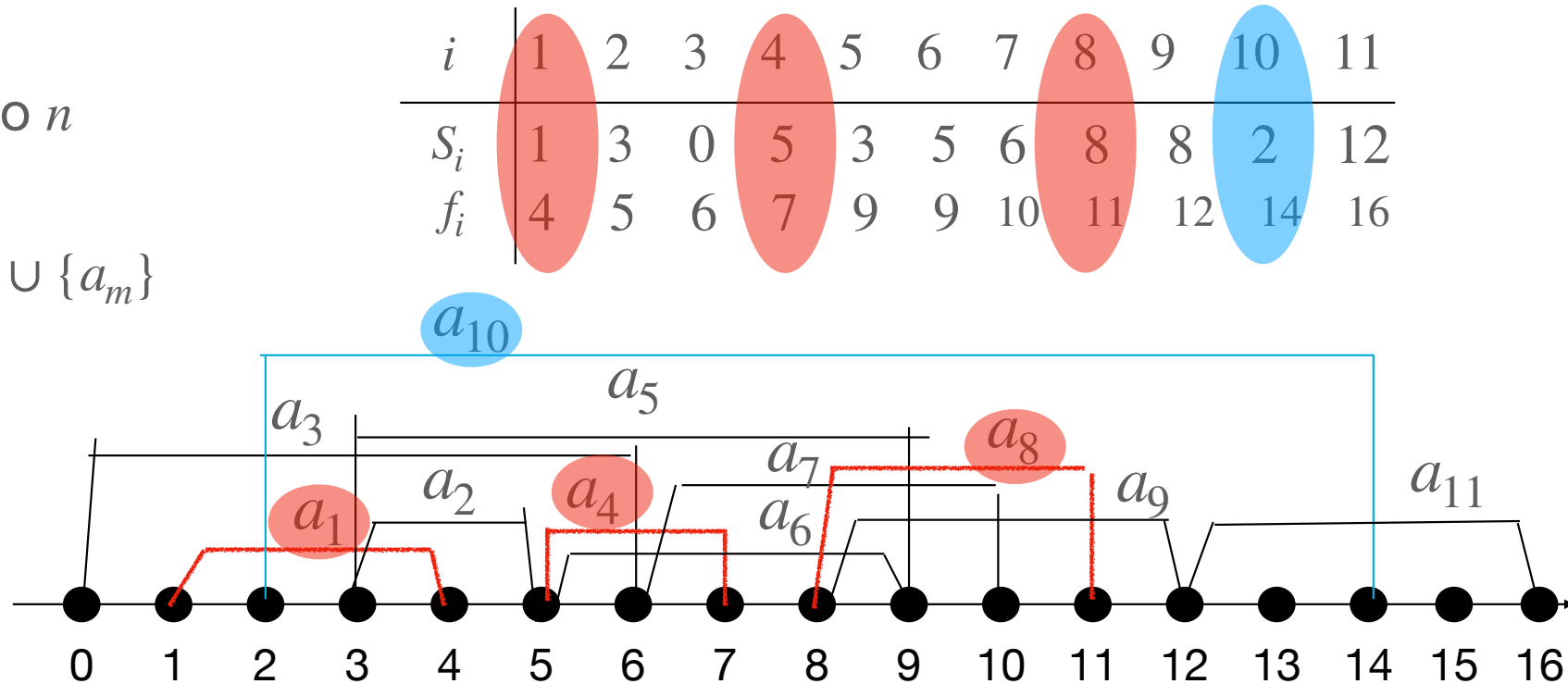
16.1 Activity Selection Problem

Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$
2. $k = 1$
3. for $m = 2$ to n
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A

$$A = \{a_1, a_4, a_8\} \quad k = 8$$



16.1 Activity Selection Problem

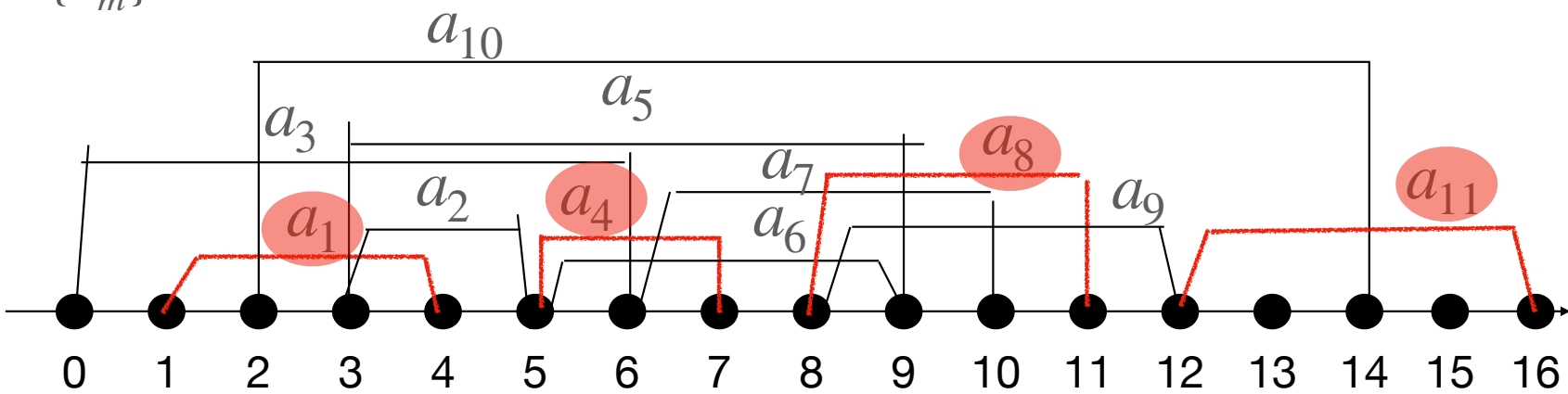
Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

- 1. $A = \{a_1\}$
- 2. $k = 1$
- 3. for $m = 2$ to n
- 4. if $s_m \geq f_k$
- 5. $A = A \cup \{a_m\}$
- 6. $k = m$
- 7. Return A

$A = \{a_1, a_4, a_8, a_{11}\}$ $k = 11$

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	9	9	10	11	12	14	16



16.1 Activity Selection Problem

Pseudo-code of the greedy algorithm:

Greedy-Activity-Selector (s,f,n):

1. $A = \{a_1\}$ $O(1)$
2. $k = 1$ $O(1)$
3. for $m = 2$ to n $O(n)$
4. if $s_m \geq f_k$
5. $A = A \cup \{a_m\}$
6. $k = m$
7. Return A $O(n)$

Time complexity: $O(n)$