Lecture 28

Activity-Selection Problem (contd.)

Activity-Selection

Activity-Selection

Act-Sel:

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output:

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Two activities a_i and a_j are mutually compatible if either $s_i \geq f_j$ or $s_j \geq f_i$.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

i	1	2	3	4	5	6	7	8	9	10	11
s_i											
f_i											

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

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

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

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

$\{a_1, a_4, a_8, a_{11}\}$ is a largest-size subset of mutually compatible activities.

Activity-Selection

Act-Sel:

Input: Given a set of n proposed activities $S = \{a_1, a_2, \ldots, a_n\}$, where each activity a_i has start time s_i and finish time f_i. If selected, activity a_i takes place during $[s_i, f_i)$.

Output: Find a largest-size subset of mutually compatible activities.

Assumption: Given activities are sorted in monotonically increasing order of finish time.

Example:

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

$\{a_1, a_4, a_8, a_{11}\}$ is a largest-size subset of mutually compatible activities.

Optimal Substructure in Activity-Selection

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

i	1	2	3	4	5	6	7	8	9	10	11
s_i											
f_i											

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

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

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

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

$S_{3,9}$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

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

$S_{3,9}$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

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

$S_{3,9}$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

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

$S_{3,9}$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

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

$S_{3,9}$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

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

$$S_{3,9} = \{a_4, a_5, a_7\}$$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

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

$$S_{5,9} = \varnothing$$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

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

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

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

For what i, j, $A_{i,j}$ will give the desired answer?

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

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

Note: We add two dummy activities a_0 and a_{n+1} at the ends so that $A_{0,n+1}$ is the desired answer.

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

Observation: $S_{i,j}$ can only contains activities from $\{a_{i+1}, a_{i+2}, \ldots, a_{j-1}\}$.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

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

Note: We add two dummy activities a_0 and a_{n+1} at the ends so that $A_{0,n+1}$ is the desired answer.

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

Optimal Substructure in Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

We will develop recurrence for $c_{i,j}$ after establishing optimal substructure

Optimal Substructure in Activity-Selection

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

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

If a_6 is part for some $A_{0,12}$, then:

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

If a_6 is part for some $A_{0,12}$, then:

- $A_{0,12} =$

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

If a_6 is part for some $A_{0,12}$, then:

- $A_{0,12} =$
- $c_{0,12} =$

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

If a_6 is part for some $A_{0,12}$, then:

- $A_{0,12} = \quad \{a_6\}$
- $c_{0,12} =$

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

If a_6 is part for some $A_{0,12}$, then:

- $A_{0,12} = A_{0,6} \cup \{a_6\}$
- $c_{0,12} =$

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

If a_6 is part for some $A_{0,12}$, then:

- $A_{0,12} = A_{0,6} \cup \{a_6\} \cup A_{6,12}$
- $c_{0,12} =$

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

If a_6 is part for some $A_{0,12}$, then:

- $A_{0,12} = A_{0,6} \cup \{a_6\} \cup A_{6,12}$
- $c_{0,12} =$

Prove by contradiction if $A_{0,12}$ that contains a_6 doesn't contain $A_{0,6}$ or $A_{6,12}$, it is not a maximum set …

Optimal Substructure in Activity-Selection

Let's try to find $A_{0,12}$ and $c_{0,12}$.

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

If a_6 is part for some $A_{0,12}$, then:

- $A_{0,12} = A_{0,6} \cup \{a_6\} \cup A_{6,12}$
- $c_{0,12} = c_{0,6} + c_{6,12} + 1$

Prove by contradiction if $A_{0,12}$ that contains a_6 doesn't contain $A_{0,6}$ or $A_{6,12}$, it is not a maximum set …

Optimal Substructure in Activity-Selection

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$ and $c_{i,j} = |A_{i,j}|$

Optimal Substructure in Activity-Selection

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$ and $c_{i,j} = |A_{i,j}|$

If $a_{i+1} \in A_{i,j}$, then $c_{i,j} = c_{i,i+1} + c_{i+1,j} + 1$

Optimal Substructure in Activity-Selection

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$ and $c_{i,j} = |A_{i,j}|$

If $a_{i+1} \in A_{i,j}$, then $c_{i,j} = c_{i,i+1} + c_{i+1,j} + 1$

If $a_{i+2} \in A_{i,j}$, then $c_{i,j} = c_{i,i+2} + c_{i+2,j} + 1$

Optimal Substructure in Activity-Selection

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$ and $c_{i,j} = |A_{i,j}|$

If $a_{i+1} \in A_{i,j}$, then $c_{i,j} = c_{i,i+1} + c_{i+1,j} + 1$

If $a_{i+2} \in A_{i,j}$, then $c_{i,j} = c_{i,i+2} + c_{i+2,j} + 1$

If $a_{i+3} \in A_{i,j}$, then $c_{i,j} = c_{i,i+3} + c_{i+3,j} + 1$

Optimal Substructure in Activity-Selection

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$ and $c_{i,j} = |A_{i,j}|$

If $a_{i+1} \in A_{i,j}$, then $c_{i,j} = c_{i,i+1} + c_{i+1,j} + 1$

If $a_{i+2} \in A_{i,j}$, then $c_{i,j} = c_{i,i+2} + c_{i+2,j} + 1$

If $a_{i+3} \in A_{i,j}$, then $c_{i,j} = c_{i,i+3} + c_{i+3,j} + 1$

$\vdots$

Optimal Substructure in Activity-Selection

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$ and $c_{i,j} = |A_{i,j}|$

If $a_{i+1} \in A_{i,j}$, then $c_{i,j} = c_{i,i+1} + c_{i+1,j} + 1$

If $a_{i+2} \in A_{i,j}$, then $c_{i,j} = c_{i,i+2} + c_{i+2,j} + 1$

If $a_{i+3} \in A_{i,j}$, then $c_{i,j} = c_{i,i+3} + c_{i+3,j} + 1$

$\vdots$

If $a_{j-1} \in A_{i,j}$, then $c_{i,j} = c_{i,j-1} + c_{j-1,j} + 1$

Optimal Substructure in Activity-Selection

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$ and $c_{i,j} = |A_{i,j}|$

If $a_{i+1} \in A_{i,j}$, then $c_{i,j} = c_{i,i+1} + c_{i+1,j} + 1$

If $a_{i+2} \in A_{i,j}$, then $c_{i,j} = c_{i,i+2} + c_{i+2,j} + 1$

If $a_{i+3} \in A_{i,j}$, then $c_{i,j} = c_{i,i+3} + c_{i+3,j} + 1$

$\vdots$

If $a_{j-1} \in A_{i,j}$, then $c_{i,j} = c_{i,j-1} + c_{j-1,j} + 1$

Optimal Substructure in Activity-Selection

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$ and $c_{i,j} = |A_{i,j}|$

If $a_{i+1} \in A_{i,j}$, then $c_{i,j} = c_{i,i+1} + c_{i+1,j} + 1$

If $a_{i+2} \in A_{i,j}$, then $c_{i,j} = c_{i,i+2} + c_{i+2,j} + 1$

If $a_{i+3} \in A_{i,j}$, then $c_{i,j} = c_{i,i+3} + c_{i+3,j} + 1$

$\vdots$

If $a_{j-1} \in A_{i,j}$, then $c_{i,j} = c_{i,j-1} + c_{j-1,j} + 1$

$c_{i,j}$ is maximum of these

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} & \\ & \end{cases}$$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} \;, & \text{if } i = j \\ & \end{cases}$$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} \quad , & \text{if } i = j \\ \quad , & \text{if } i \neq j \end{cases}$$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} \quad , & \text{if } S_{i,j} = \varnothing \\ \quad , & \text{if } S_{i,j} \neq \varnothing \end{cases}$$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \emptyset \\ \qquad\qquad\qquad\qquad, & \text{if } S_{i,j} \neq \emptyset \end{cases}$$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \varnothing \\ max_{k \in \{i+1, j-1\}}(c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \varnothing \end{cases}$$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \varnothing \\ max_{k \in \{i+1, j-1\}}(c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \varnothing \end{cases}$$

↖

Before including an activity in $A_{i,j}$ we should check whether it is in $S_{i,j}$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \emptyset \\ \max_{k \in \{i+1, j-1\}} (c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \emptyset \end{cases}$$

Before including an activity in $A_{i,j}$ we should check whether it is in $S_{i,j}$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \emptyset \\ max_{a_k \in S_{i,j}}(c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \emptyset \end{cases}$$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \emptyset \\ max_{a_k \in S_{i,j}}(c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \emptyset \end{cases}$$

Implementing DP will now take $O(n^3)$ time.

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \emptyset \\ max_{a_k \in S_{i,j}}(c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \emptyset \end{cases}$$

↑

Implementing DP will now take $O(n^3)$ time. Can we do better?

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \emptyset \\ max_{a_k \in S_{i,j}}(c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \emptyset \end{cases}$$

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \emptyset \\ max_{a_k \in S_{i,j}}(c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \emptyset \end{cases}$$

↑

We do NOT need go over all $a_k \in S_{i,j}$.

Recurrence for Activity-Selection

$S_{i,j}$ = Set of activities that start after activity a_i finishes and that finish before activity a_j starts.

$A_{i,j}$ = A maximum set of mutually compatible activities in $S_{i,j}$

$c_{i,j} = |A_{i,j}|$

$$c_{i,j} = \begin{cases} 0, & \text{if } S_{i,j} = \emptyset \\ max_{a_k \in S_{i,j}}(c_{i,k} + c_{k,j} + 1), & \text{if } S_{i,j} \neq \emptyset \end{cases}$$

↑

We do NOT need go over all $a_k \in S_{i,j}$. We can make a GREEDY choice.

Greedy Choice for Activity-Selection

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Some possible intuitive choices:

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Some possible intuitive choices:

- Activity that finishes first.

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Some possible intuitive choices:

- Activity that finishes first.
- Activity that starts last.

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Some possible intuitive choices:

- Activity that finishes first.
- Activity that starts last.
- Activity with shortest duration.

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Some possible intuitive choices:

- Activity that finishes first. ✓
- Activity that starts last.
- Activity with shortest duration.

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Some possible intuitive choices:

- Activity that finishes first. ✓
- Activity that starts last. ✓
- Activity with shortest duration.

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Some possible intuitive choices:

- Activity that finishes first. ✓
- Activity that starts last. ✓
- Activity with shortest duration. ✗

Greedy Choice for Activity-Selection

Which activity is definitely a part of $A_{i,j}$?

Intuition says an activity which leaves the time for as many other activities as possible.

Some possible intuitive choices:

- Activity that finishes first. ✓ ← We will design a greedy algorithm based on this.
- Activity that starts last. ✓
- Activity with shortest duration. ✗

Proving Correctness of Greedy Choice

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

Proving Correctness of Greedy Choice

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

Why a_1 will definitely be a part of some $A_{0,12}$?

Proving Correctness of Greedy Choice

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

Why a_1 will definitely be a part of some $A_{0,12}$?

Consider an $A_{0,12}$, say $\{a_2, a_4, a_9, a_{11}\}$, not containing a_1.

Proving Correctness of Greedy Choice

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

Why a_1 will definitely be a part of some $A_{0,12}$?

Consider an $A_{0,12}$, say $\{a_2, a_4, a_9, a_{11}\}$, not containing a_1.

But a_2 can be replaced with a_1 to produce another $A_{0,12}$.

Proving Correctness of Greedy Choice

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$,

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

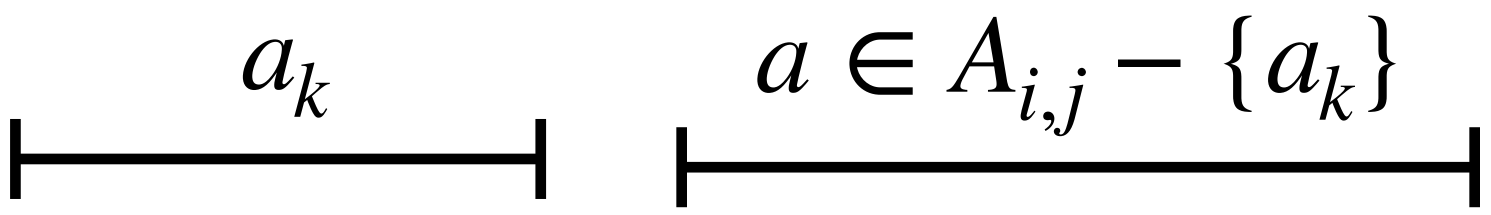
Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.



Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

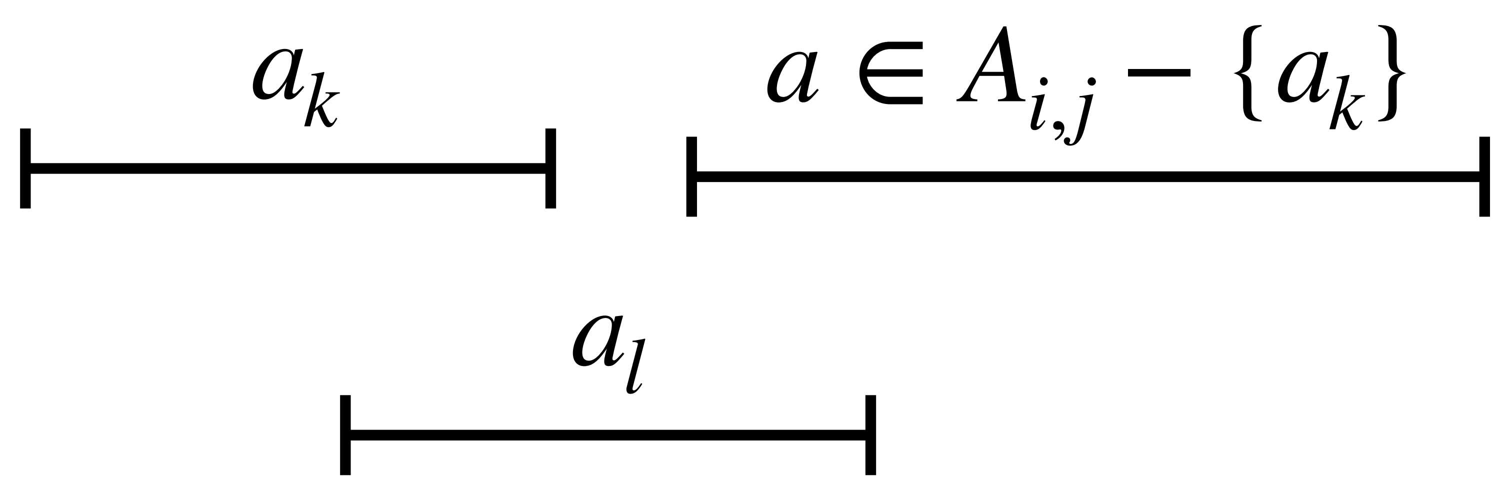
Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.



Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

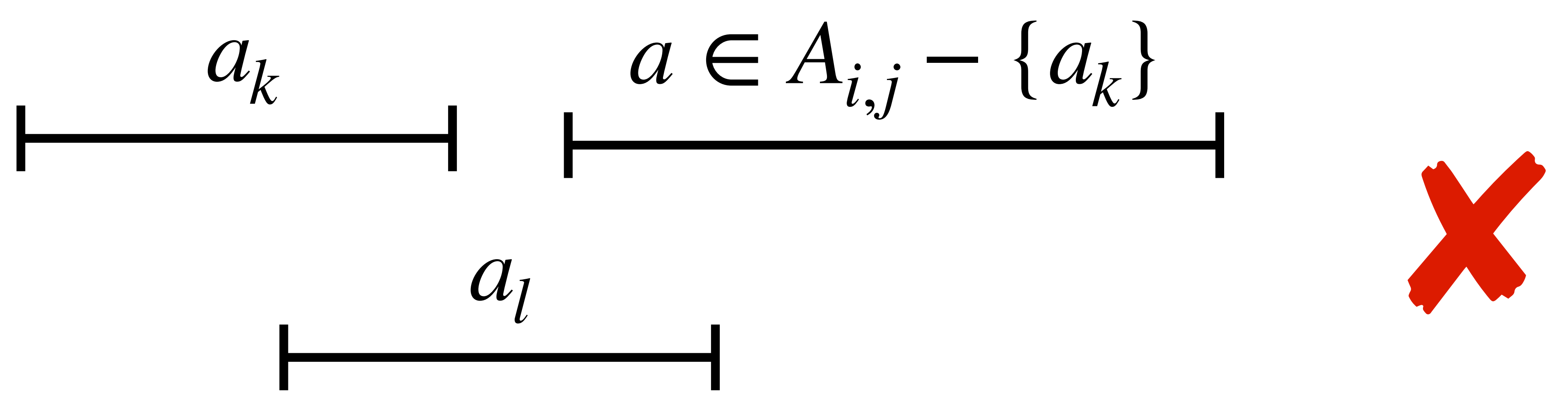
Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.



Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

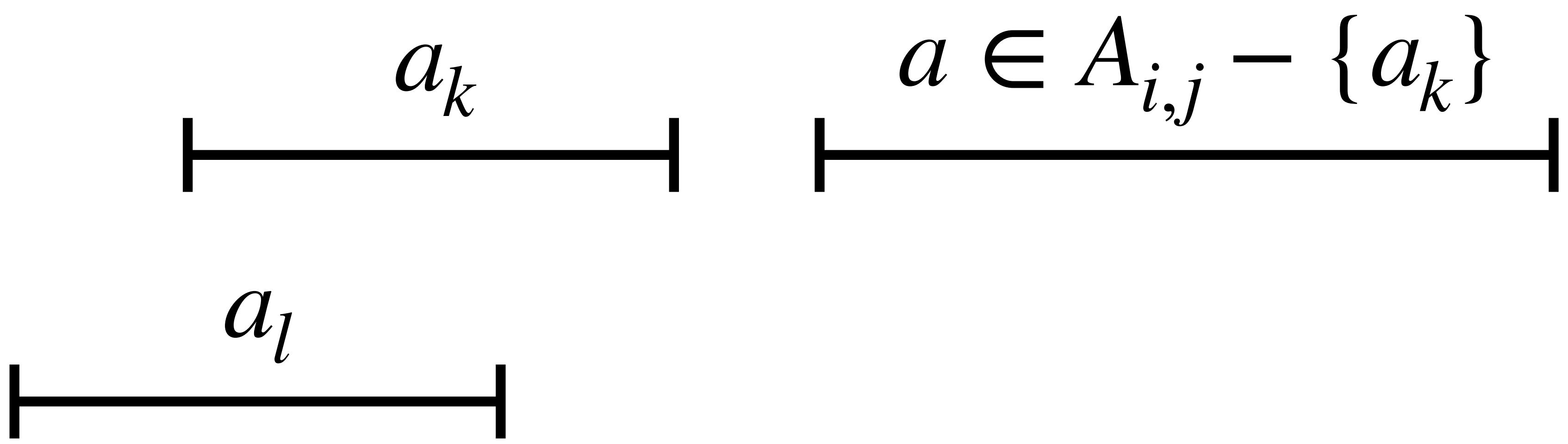
Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.



Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

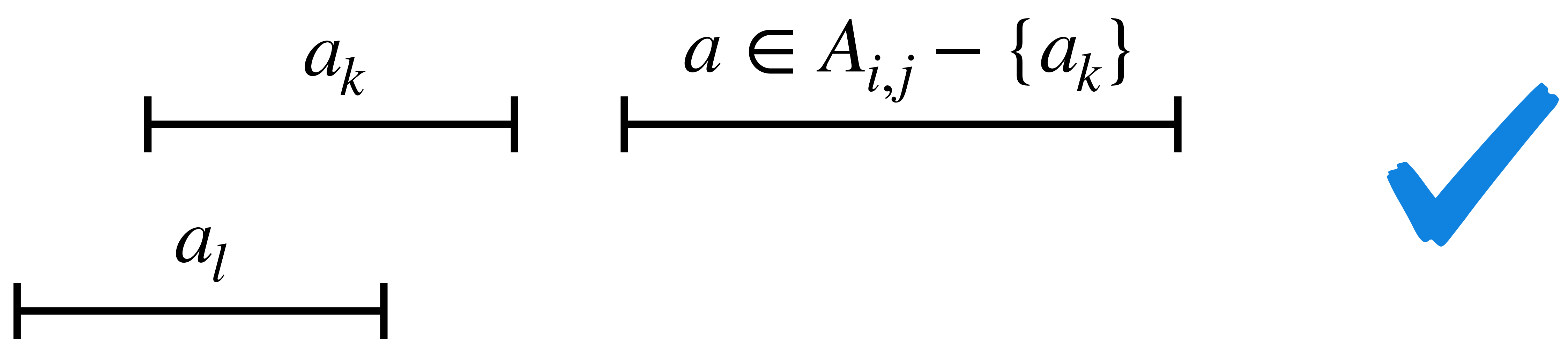
Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.



Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.
- Since a_l finishes "before" a_k, activities in $A_{i,j} - \{a_k\}$ will start after a_l as well.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.
- Since a_l finishes "before" a_k, activities in $A_{i,j} - \{a_k\}$ will start after a_l as well.

Clearly, $|A'| = |A_{i,j}|$. Hence, proved.

Proving Correctness of Greedy Choice

Lemma: Earliest finishing activity in $S_{i,j}$ will be part of some $A_{i,j}$.

Proof: Let a_l be an activity with earliest finish time in $S_{i,j}$.

Consider some $A_{i,j}$. If $a_l \in A_{i,j}$, we are done.

If $a_l \notin A_{i,j}$, consider $A' = A_{i,j} - \{a_k\} + \{a_l\}$, where a_k is the earliest finishing activity in $A_{i,j}$.

We claim that A' is also a set of mutually compatible activities:

- a_k has earliest finish time in $A_{i,j}$, hence activities in $A_{i,j} - \{a_k\}$ will start after a_k.
- Since a_l finishes "before" a_k, activities in $A_{i,j} - \{a_k\}$ will start after a_l as well.

Clearly, $|A'| = |A_{i,j}|$. Hence, proved. ■