Dept. of CSE
Varendra University

Greedy Algorithms:
Activity Selection Problem

Course Instructor:

Sumaiya Tasnim

Lecturer, Department of CSE

Varendra University

Acknowledgement

Mosiur Rahman Sweet
Former Lecturer, Department of CSE
Varendra University

Md. Muktar Hossain
Lecturer, Department of CSE
Varendra University

Activity Selection Problem

- It is a problem of scheduling several activities that require exclusive use of a common resource, with the goal of selecting a maximum size set of **mutually compatible activities.**

Activity	1	2	3	4	5	6	7
Start	1	3	5	2	6	8	9
Finish	4	6	7	9	10	11	11
After sorting by finish time in ascending order							

Activity Selection Problem

Activity	1		3			6	
Start	1		5			8	
Finish	4		7			11	

	1	2	3	4	5	6	7	8	9	10	11
1	🟩	🟩	🟩	🟩							
2			🟨	🟨	🟨	🟨					
3					🟩	🟩	🟩				
4		🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	
5						🟦	🟦	🟦	🟦	🟦	
6								🟩	🟩	🟩	🟩
7									⬛	⬛	⬛

Activity Selection Algorithm

- Sort the activities by their finishing times (in ascending order).
- Select the first activity (which finishes the earliest).
- Iterate through the remaining activities:
  - If the start time of the current activity is greater than or equal to the finish time of the last selected activity, select it.
- Continue until all activities are checked.

Problem Statement

Given ***n*** activities with their ***start*** and ***finish*** times, the goal is to select the ***maximum number of activities*** that can be performed by a single person, assuming that the person can work on only one activity at a time.

Input

- An integer **N** representing the number of activities.
- An array **start[ ]** of size **N** where **start[i]** is the start time of the **i-th** activity.
- An array **finish[ ]** of size **N** where **finish[i]** is the finish time of the **i-th** activity.

Objective

- Select the maximum number of activities that can be performed without any overlapping.

Constraints

- $1 \leq N \leq 10^5$ (large input sizes possible)
- $1 \leq$ ***start[i], finish[i]*** $\leq 10^9$

Pseudocode:

```
ACTIVITY_SELECTION(start[], finish[], N)
    Sort activities based on finish times
    Select the first activity and print it
    lastSelected = 0

    FOR i = 1 to N-1 DO
        IF start[i] >= finish[lastSelected] THEN
            Select activity i and print it
            lastSelected = i
        ENDIF
    END FOR
END
```

Time Complexity

- **Sorting the activities**: O(nlogn)
- **Iterating through activities**: O(n)
- **Overall Complexity**: O(nlogn)

An Example

A person is given a set of activities, each with a specific start and finish time.
The goal is to select the maximum number of non-overlapping activities that can be performed by a single person.

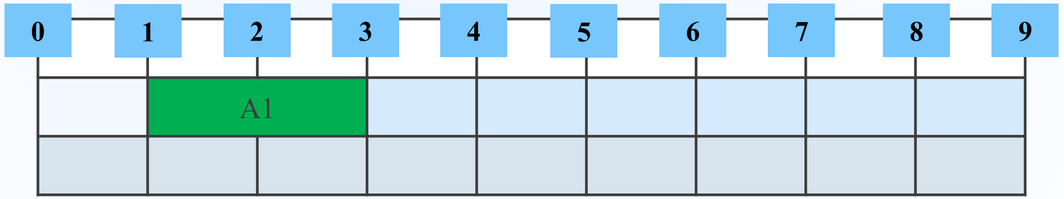
Activity	Start Time	Finish Time
1	1	3
2	2	5
3	4	6
4	6	8
5	5	7
6	8	9

Sort activities based on finish times

Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9

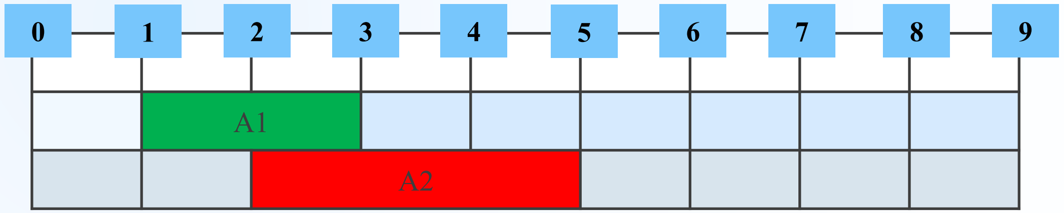
Select the first activity and print it
lastSelected = 0

Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9



```
FOR i = 1 to N-1 DO
    IF start[i] >= finish[lastSelected] THEN
        Select activity i and print it
        lastSelected = i
    ENDIF
END FOR
```

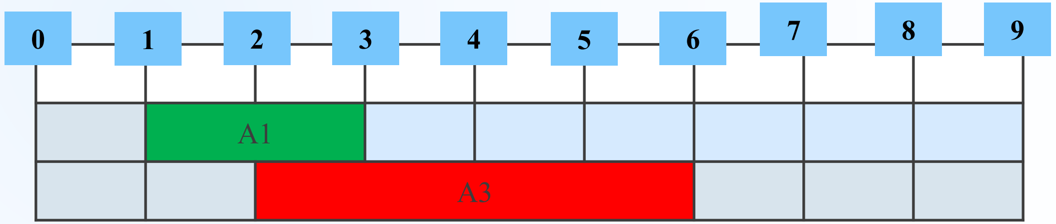
Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9



Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9

```
FOR i = 1 to N-1 DO
    IF start[i] >= finish[lastSelected] THEN
        Select activity i and print it
        lastSelected = i
    ENDIF
END FOR
```

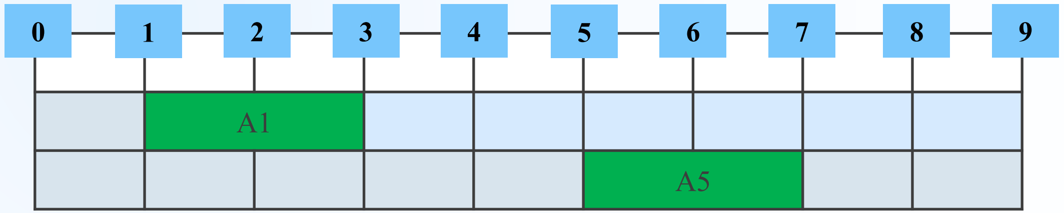
Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9



Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9

Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9

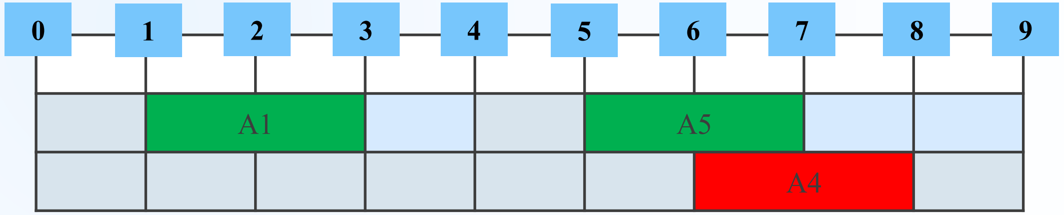
```
FOR i = 1 to N-1 DO
    IF start[i] >= finish[lastSelected] THEN
        Select activity i and print it
        lastSelected = i
    ENDIF
END FOR
```



Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9

Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9

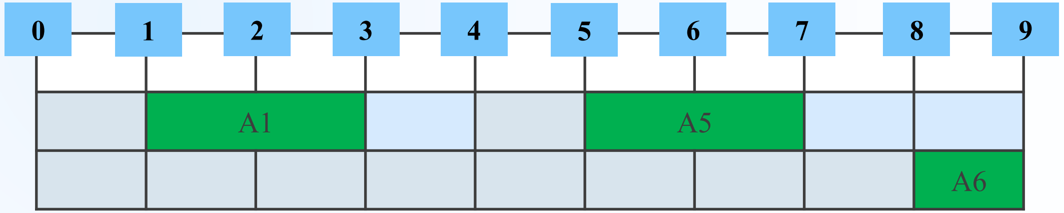
```
FOR i = 1 to N-1 DO
      IF start[i] >= finish[lastSelected] THEN
         Select activity i and print it
         lastSelected = i
      ENDIF
   END FOR
```



Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9

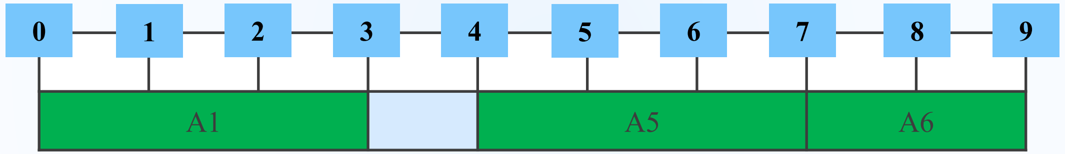
```
FOR i = 1 to N-1 DO
    IF start[i] >= finish[lastSelected] THEN
        Select activity i and print it
        lastSelected = i
    ENDIF
END FOR
```

Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9



Activity	1	2	3	5	4	6
Start Time	1	2	2	5	6	8
Finish Time	3	5	6	7	8	9

Example Cont'd



Applications

- **Job scheduling:** Scheduling jobs or tasks where each task has a deadline.
- **Conference room scheduling:** Allocating meeting rooms based on time slots.
- **CPU scheduling:** Selecting processes that can run in a non-overlapping manner.
- **Event planning:** Scheduling events with time constraints.

Thank You