Activity Selection Problem

Mashiwat Tabassum Waishy

Lecturer

Department *of*
CSE
STAMFORD UNIVERSITY BANGLADESH

Activity Selection Problem

- An Activity Selection Problem is the problem of scheduling a resource among several competing activity.

Activity Selection

- The activity scheduling is a simple scheduling problem fro which the greedy algorithm approach provides an optimal solution.
- We are given a set $S = \{a_1, a_2, \ldots \ldots, a_n\}$ of n activities that are to be scheduled to use some resources.
- Each activity a_i must be started at a given start time s_i and ends at a given finish time f_i.
- Example:
  - An Example is that a number of lectures are to be given in a single lecture hall.
  - The start and end times have be set up in advance.
  - The lectures are to be scheduled.
  - There is only one resource(e.g. lecture hall)
  - Some start and finish times may overlap.
  - Therefore, not all request can be honored.

Activity Selection

- We sat that two activities a_i and a_j are non-interfering if their start-finish intervals do not overlap.
- i.e. $(s_i, f_i) \cap (s_j, f_j) = \emptyset$.
- The activity selection problem is to select a maximum-size set of mutually non-interfering activities for the resource.

Activity Selection Problem

- A set s={1,2…n} proposed activities.
- Everyone want the same resource (as like classroom) used by one at a time.
- Each activity ‘i’ has start time s_i and finish time f_i where $s_i <= f_i$. Time between [s_i and f_i] is the i^{th} activity time.
- Two activities i and j are compatible if they do not overlap ($s_i >= f_j$ or $s_j >= f_i$).

Example

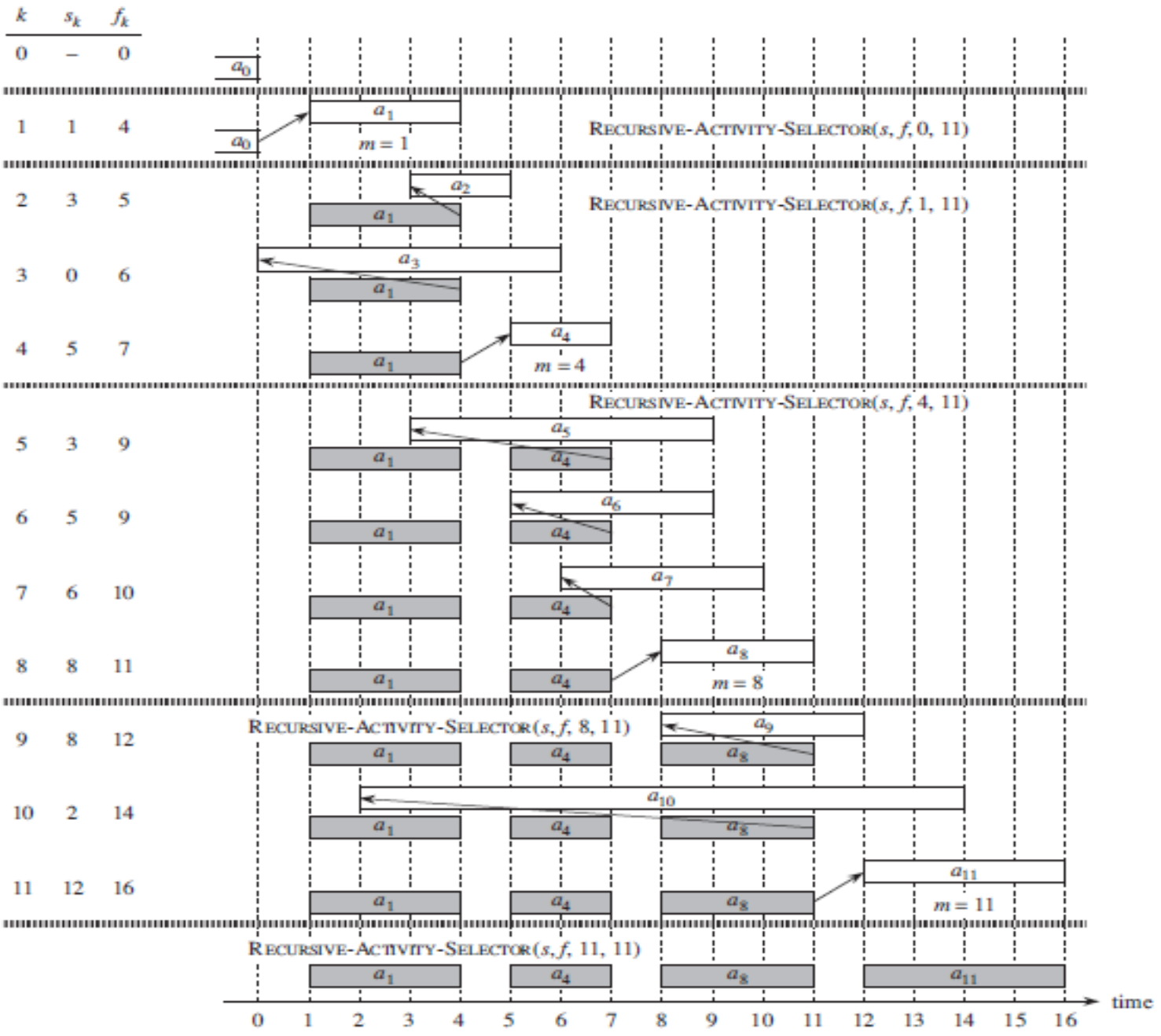
i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	9	9	10	11	12	14	16

The largest subset of mutually compatible activities:

{a1; a4; a8; a11}

Algorithm

```
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 finish
3      m = m + 1
4  if m ≤ n
5      return {a_m} ∪ RECURSIVE-ACTIVITY-SELECTOR(s, f, m, n)
6  else return ∅
```


k
s_k
f_k
0 – 0
1 1 4
2 3 5
3 0 6
4 5 7
5 3 9
6 5 9
7 6 10
8 8 11
9 8 12
10 2 14
11 12 16
a_0
a_1
a_2
a_3
a_4
a_5
a_6
a_7
a_8
a_9
a_{10}
a_{11}
$m = 1$
$m = 4$
$m = 8$
$m = 11$
RECURSIVE-ACTIVITY-SELECTOR(s, f, 0, 11)
RECURSIVE-ACTIVITY-SELECTOR(s, f, 1, 11)
RECURSIVE-ACTIVITY-SELECTOR(s, f, 4, 11)
RECURSIVE-ACTIVITY-SELECTOR(s, f, 8, 11)
RECURSIVE-ACTIVITY-SELECTOR(s, f, 11, 11)
time
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Activity Selection Problem – Greedy Solution

1. Sort the input activities by increasing finishing time as $f_1 <= f_2 <= \ldots.. <= f_n$.
2. Now, define or call a procedure GREEDY- ACTIVITY-SELECTOR (s,f)

Greedy Algorithm

```
GREEDY-ACTIVITY-SELECTOR(s, f)
1  n = s.length
2  A = {a_1}
3  k = 1
4  for m = 2 to n
5      if s[m] ≥ f[k]
6          A = A ∪ {a_m}
7          k = m
8  return A
```

Analysis

1. Sort input activity by increasing finishing time requires O(n log n).
2. Greedy-activity-selector (s,f) requires $\Theta(n)$.


Thank
you