1/12/2023 — LECTURE NO.14 — Friday

[redacted]

- ACTIVITY SELECTION:

Given activities ~~and~~ (lectures) and one room. Now schedule as many activities as possible such that they do not conflict each other.

1. Min Duration:

i	1	2	3	4	5	6	7	8	9
Si	2	2	4	1	5	8	9	11	13
Fi	3	5	7	8	9	10	11	14	16
di	1	3	3	7	4	2	2	3	3
	✓	X	~~X~~ ✓	X	X	✓	X	✓	X

Solution = $\{a_1, a_3, a_6, a_8\}$ → optimal. sol.

i	1	2	3
Si	4	1	5
Fi	6	5	10
di	2	4	5
	✓	X	X

Greedy Sol. = $\{a_1\}$

But

optimal Sol = $\{a_2, a_3\}$

So this approach does not give optimal Solution at each instance (every set of activities)

2. Min Start Time:

We cannot choose this approach also because there is possibility that the activity starting at The minimum time may end at tha last which will result in selection of only one activity. That will also ~~not~~ be a non-optimal solution.

3. Min End Time:

i	1	2	3	4	5	6	7	8	9
s_i	2	2	4	1	5	8	9	11	13
f_i	3	5	7	8	9	10	11	14	16
	✓	X	✓	X	X	✓	X	✓	X

Solution = $\{a_1, a_3, a_6, a_8\}$ → Optimal Solution.

i	1	2	3
s_i	4	1	5
f_i	6	5	10
	X	✓	✓

Solution = $\{a_2, a_3\}$ → Optimal Solution

4. Max Start Time.

i	1	2	3	4	5	6	7	8	9
Si	2	2	4	1	5	8	9	11	13
Fi	3	5	7	8	9	10	11	14	16
	X	✓	X	X	~~X~~ ✓	X	✓	X	✓

Solution = $\{a_2, a_5, a_7, a_9\}$ → optimal

i	1	2	3
Si	4	1	5
Fi	6	5	10
	X	✓	✓

Solution = $\{a_2, a_3\}$ → optimal.

1) Min duration X
2) Min Start Time X
3) Min End Time ✓ }
4) Max Start Time. ✓ } optimal Solution at every instance.

Activity Selection Algorithm.

```
                Arrays.    No. of Activities
Greedy A1( S, F, n )
{  // Sort activities in Both arrays by finish
   time in ~~de~~ ascending order. → (nlogn)
   add[0] to A.
   for(i=1 to i=n-1) → n.
   [ k=1
     if S[i] ≥ F[k]
       Add A[i] to A
       k=i  }
```

Time Complexity:
$= n + n\log n$
$= O(n\log n)$.