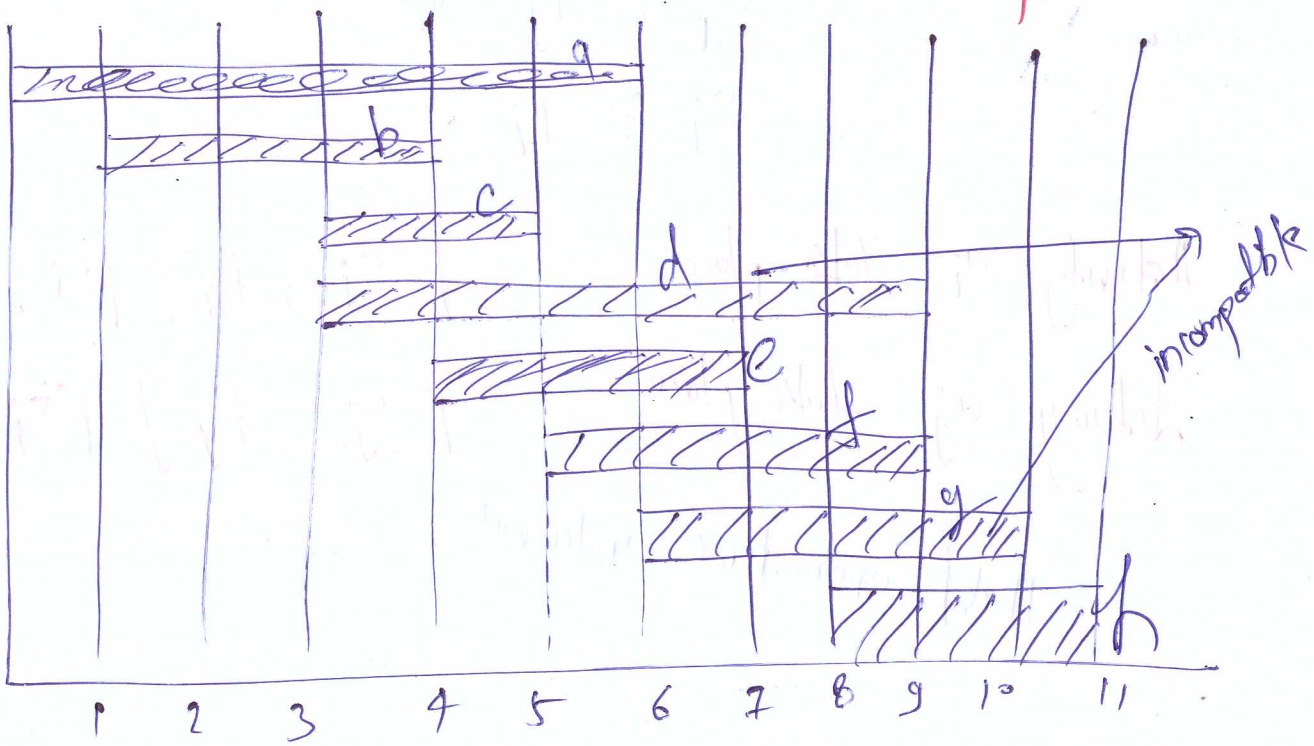
Greedy Algo

- Works well on optimization problem with following charact —
  - ↳ Make decision incrementaly in small steps without BACKTRACKING
  - ↳ Decision at each step is based on improving local state
  - ↳ Decisions are based on some fixed / simple priority RULES

Job scheduling / INTERVAL SCHEDULING / Activity Selection problem



Given One Resource

* Job j starts at s_j
  Finishes at f_j

* Two jobs are Compatible if they Do not OVERLAP

* GOAL Find the maximum subset of mutually Compatible intervals.

d and g are incompatible.

Approach

$$0 \leq s_j < f_j < \infty$$

Two jobs are compatible iff (ith and jth jobs)

$$s_i \geq f_j \text{ or}$$
$$s_j \geq f_i$$

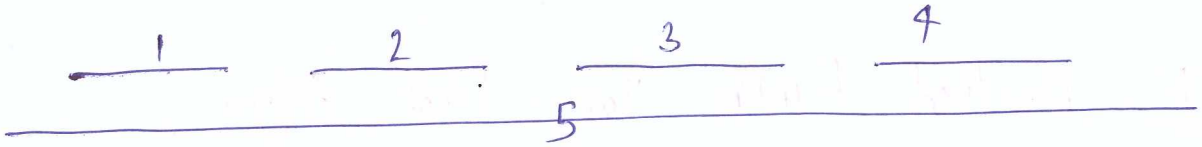
Activity a_i Take place $[s_i, f_i]$ ~~$[s_i, f_j)$~~

Activity a_j take place ~~$[s_j, f_j]$~~ ~~$[s_j, f_j)$~~

Half open time interval

- Consider interval in some natural order. Take each interval provided its compatible with one already taken

- **Earliest Start Time** Consider interval in ascendy order of S_j



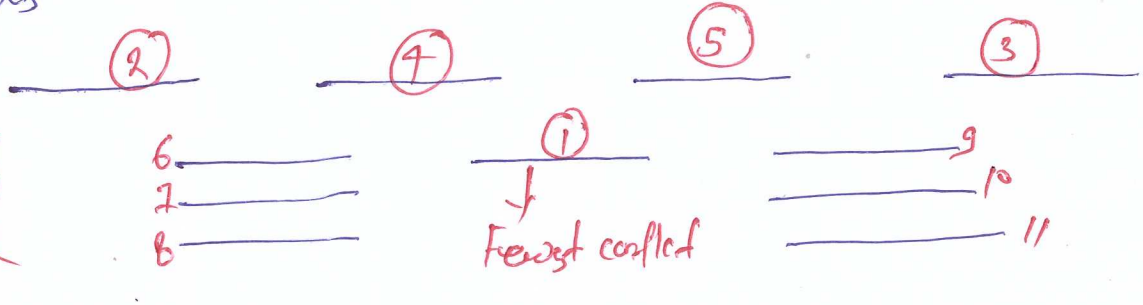
Will pick 5 as it has EST so not optimal

- **Earliest Finish Time** Ascending order of S_j

- **Shortest Interval** Minimum $f_j - S_j$

3 is shortest interval and not optimal

- **Fewest Conflicts** For each interval, count the no of conflicting intervals and pick one which has min. conflicts intervals.



Greedy Interval scheduling

- Use a simple Rule to select i
- Reject all requests incompatible with i
- Repeat UNTILL all are processed.

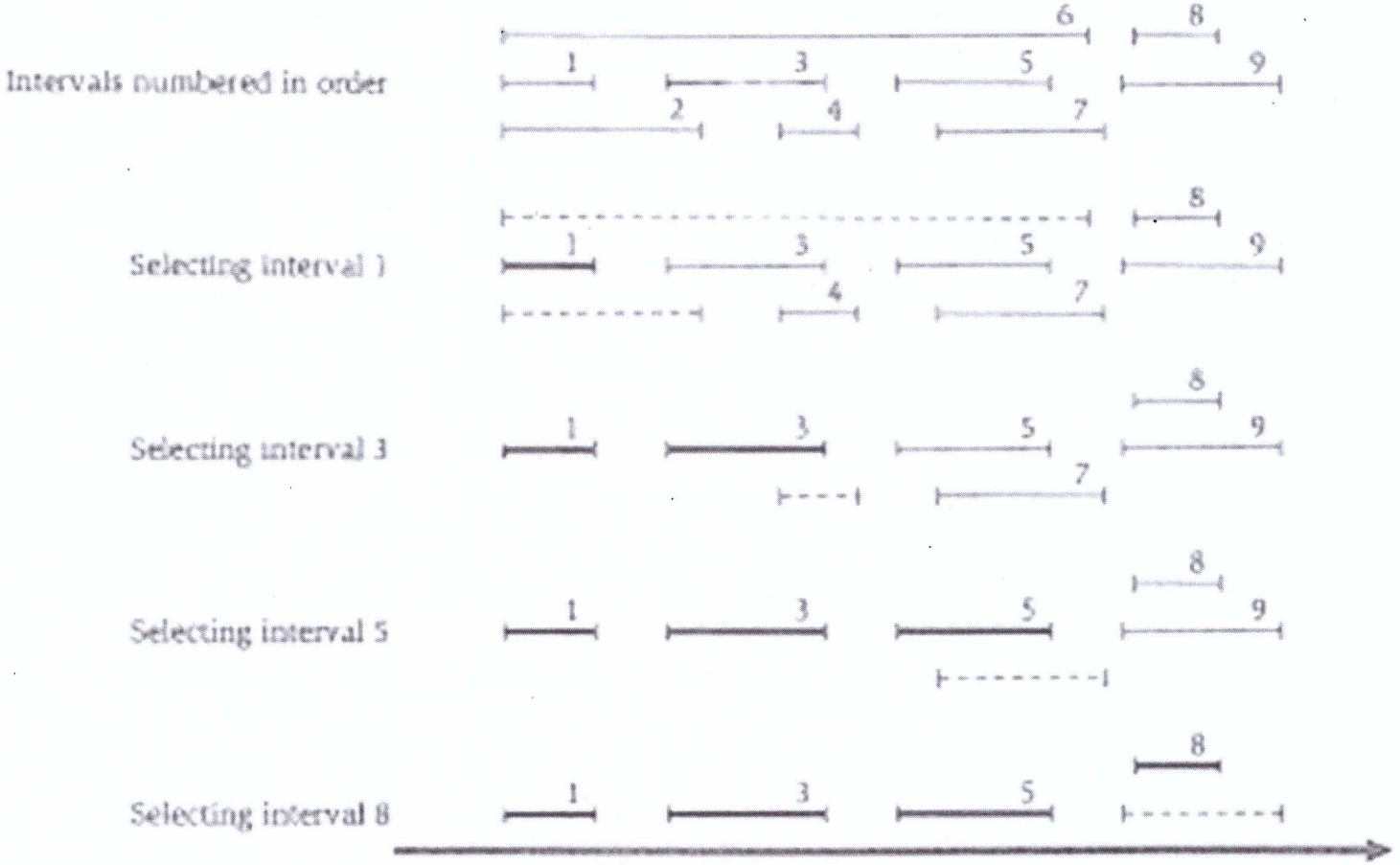
EFTF Earliest Finish Time First Algo.

1) Sort jobs in increasing (non-decreasing) order of f_j so that $f_1 \leq f_2 \leq \dots \leq f_n$ $O(n \log n)$
2) $S \leftarrow \phi$ // NULL set $O(1)$
3) For $j \leftarrow 1$ to n — $O(n)$
   - If job j is compatible with S — $O(1)$
     - $S \leftarrow S \cup \{j\}$
4) Return S

How

By tracking the finishing time of most Recently scheduled interval, this can be done in $O(1)$ time.

Sample run of Interval scheduling Algorithm



Claim 1 The algorithm outputs a list of interval

$\langle (s_1, f_1) \quad (s_2, f_2), \text{---} \quad (s_n, f_n) \rangle$

such that

$$s_1 < f_1 \leq s_2 < f_2 \leq \text{---} \leq s_n < f_n$$

Proof By Contradiction

If $f_j > s_{j+1}$ then j and $j+1$ intersect and this contradiction.

Claim 2 Given a list of intervals, greedy algorithm with Earliest Finish Time produces optimal interval.

Proof

Let we have an optimal solution

$O = \{ j_1 \; j_2 \; j_3 \; \text{---} \; j_m \}$

Solution by Greedy Algorithm

$A = \{ i_1 \; i_2 \; i_3 \; \text{---} \; i_k \}$

* Assume all the intervals in O and A are sorted by the time of interval
* We have to prove $|O| = |A|$ i.e. $m = k$

* $A = \{ i_1 \; i_2 \; \cdots \; i_k \}$

Assume sorted in finishing time

$f(i_1) < f(i_2) < \cdots < f(i_k)$ /

$f(i_1) \leq s(i_2)$, $f(i_2) \leq s(i_3)$

* $O = \{ j_1 \; j_2 \; \text{——} \; j_m \}$ OPTIMAL

Assume sorted

$f(j_1) < f(j_2) \longleftarrow < f(j_m)$

$f(j_1) \leq s(j_2)$, $f(j_2) \leq s(j_3)$, ..

SHOW $\boxed{K = m}$

Claim For each $r \leq k$ $f(i_r) \leq f(j_r)$

* Meaning greedy solution remain Ahead of Optimal solution

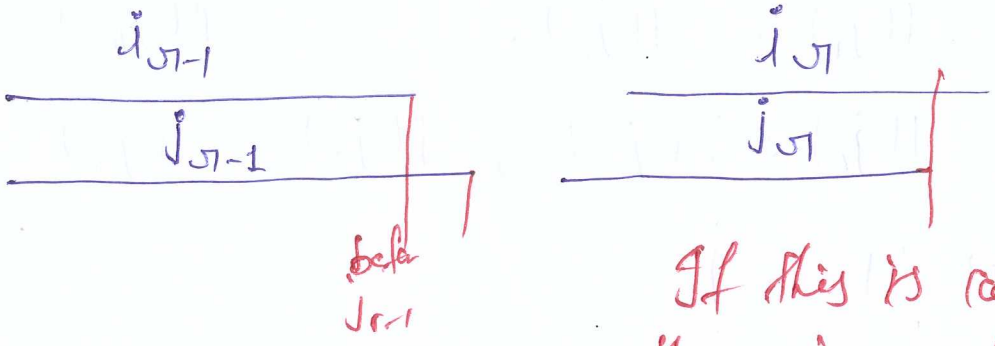
PROOF Induction on r

BASE $r = 1$

i_1 is the earlist finish time among all jobs so it must be less than $f(j_1)$ as $f(j_1)$ cannot be less than overall minimum

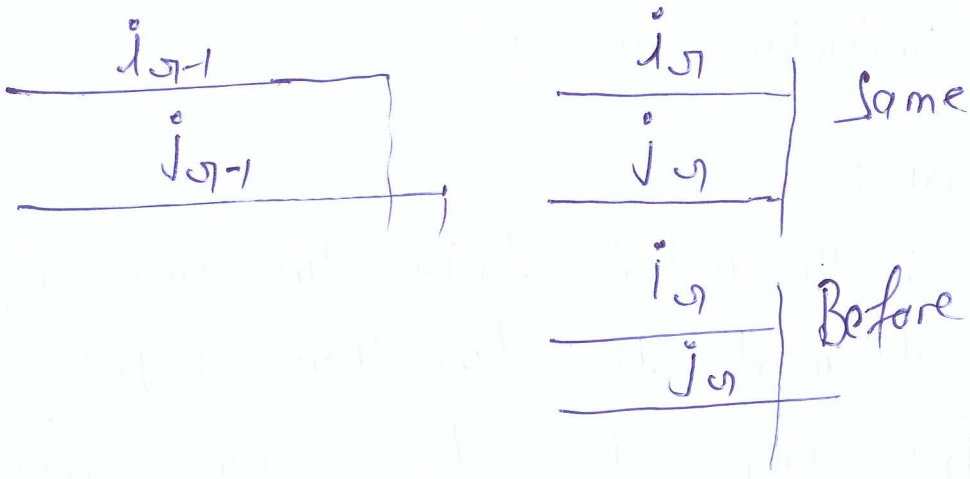
Inductive Step

- $r > 1$, Assume by induction that
$$f(i_{r-1}) \leq f(j_{r-1})$$
- Then it must be the case that $f(i_r) \leq f(j_r)$
- If not the algorithm would choose j_r rather than i_r



If this is case then j_r would be picked

To pick i_r, it should end before or at same time than j_r i.e.



* Now prove solution is optimal

Intitively we have proved that Greedy remains AHEAD.

→ Let $m > K$

→ We know that $f(i_K) \leq f(j_K)$

→ Consider j_{K+1} in O

- Greedy algo. terminates when the set becomes empty
- Since $f(i_K) \leq f(j_K) < s(j_{K+1})$

this booking is compatible with A

$= \{ i_1 \; i_2 \; \text{---} \; i_K$

→ After selecting i_K, the set still contains j_{K+1} Contradiction

i_K is compatible with j_{K+1} so it means Set is not empty.