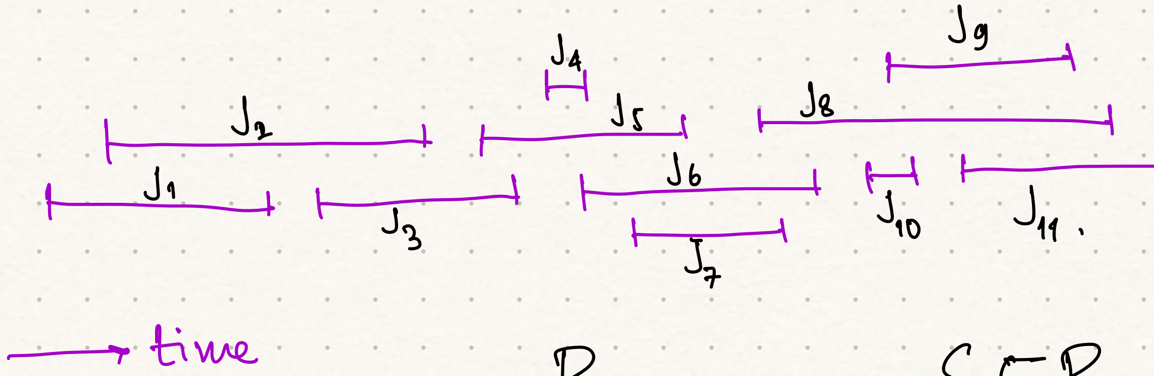
Interval Scheduling

Premise: • We have <u>one</u> processor and a set of requests/jobs

• We are given the set of requests as a list of intervals.

• We say a subset of jobs as compatible if they do not have an overlap "pairwise".

{All jobs have same priority



R, $S \subseteq R$ s.t. $\forall j_1, j_2,\ I_{j_1} \cap I_{j_2} = \phi$

Want: Largest no. of compatible jobs.

• Shortest time to completion / $\boxed{\text{Early finish times.}}$ ✓

✗ • Early start time. (FCFS)

✗ • Shortest time intervals.

✗ • Fewer incompatibilities.

Algo(R):

$A \leftarrow \phi$

While R is not empty:

• Choose a req i w/ least finish time.

• $A \leftarrow A \cup \{i\}$

- Remove all jobs that are incompatible with i from R along with i.

Return A.

$|A|$ $\quad$ k

Correctness:

The algo returns the subset $A = \{I_1, \cdots, I_k\}$. For the sake of contradiction, let us assume that there is a set $O = \{J_1, \cdots, J_m\}$ s.t. $m > k$.

$f(I) \leftarrow$ denotes the finish time of I.

Claim: $f(I_1) \leq f(J_1)$.

$O = \{J_1, \cdots, J_k, \cdots, J_m\}$
$A = \{I_1, \cdots, I_k\}$.

Lemma: $\forall\, r \leq k \quad f(I_r) \leq f(J_r)$.

$J_{k+1} \cdots J_m$ are compatible w/ A and this contradicts the termination condition of Algo.

Proof by induction on $r \in [1, \cdots, k]$.

- Base case is given by the claim. For an arbitrary $r \in [2, \cdots, k]$.
- I.H: $f(I_{r-1}) \leq f(J_{r-1})$.
- I.S:

After the $(r-1)^{th}$ job finishes in A, it picks a job (I_r) with earlier finish time.

Then $f(I_r) \leq f(J^*)$ $\forall\, J^*$ compatible with I_{r-1} in the updated set R.

In fact J_r is one such job.

In other words if $f(J_r) < f(I_r)$ then algo must choose J_r instead of I_r. But algo chose $I \Rightarrow f(I_r) \leq f(J_r)$.

$\Rightarrow f(I_r) \leq f(J_r)$.