Lecture 4 (01/30)

Greedy algorithms [Section 4.1]

→ Algorithm progresses by making 'local greedy choices'
(optimization problem)

→ Often easy to specify (ie give a greedy algorithm)

Harder part: proving greedy works!

① Interval Scheduling

Input: A sequence of n requests

Each request i: $s(i) \rightarrow$ start time
$f(i) \rightarrow$ finish time.

$(s(1), f(1)), (s(2), f(2)), \ldots, (s(n), f(n))$

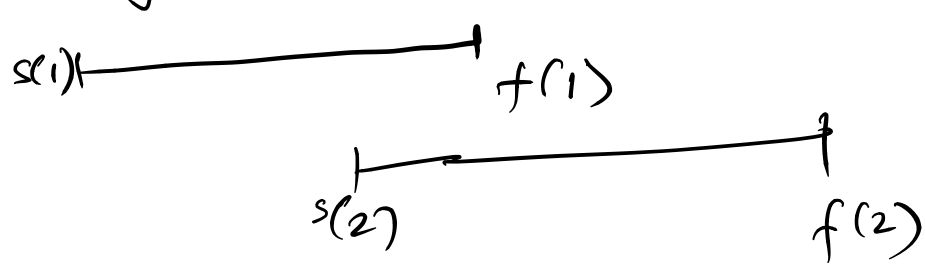
$(s(i) < f(i))$

examples: scheduling processes on a server, lectures in a particular classroom

Def^n: Compatible Sheedule: A set of requests that do not overlap.

Size of a schedule: # of requests in it.

overlapping jobs



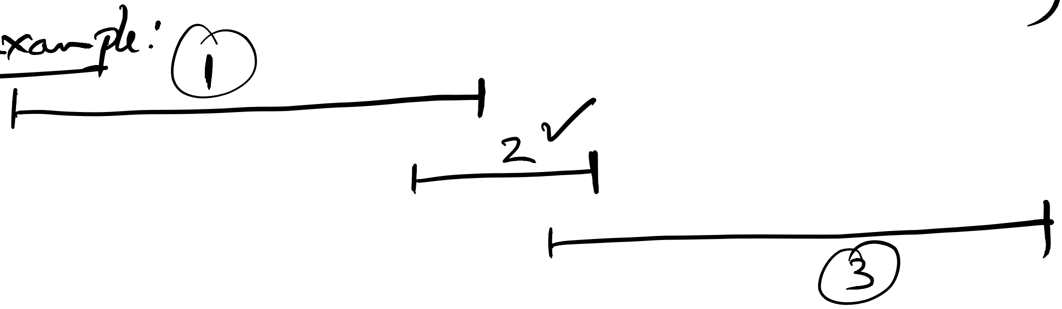
Task: Output a maximal sized compatible schedule.

Suggestions for algorithms.

Attempt 1: (a) Sort jobs/requests by time lengths.

(b) Schedule compatible jobs in ↑ (increasing order)

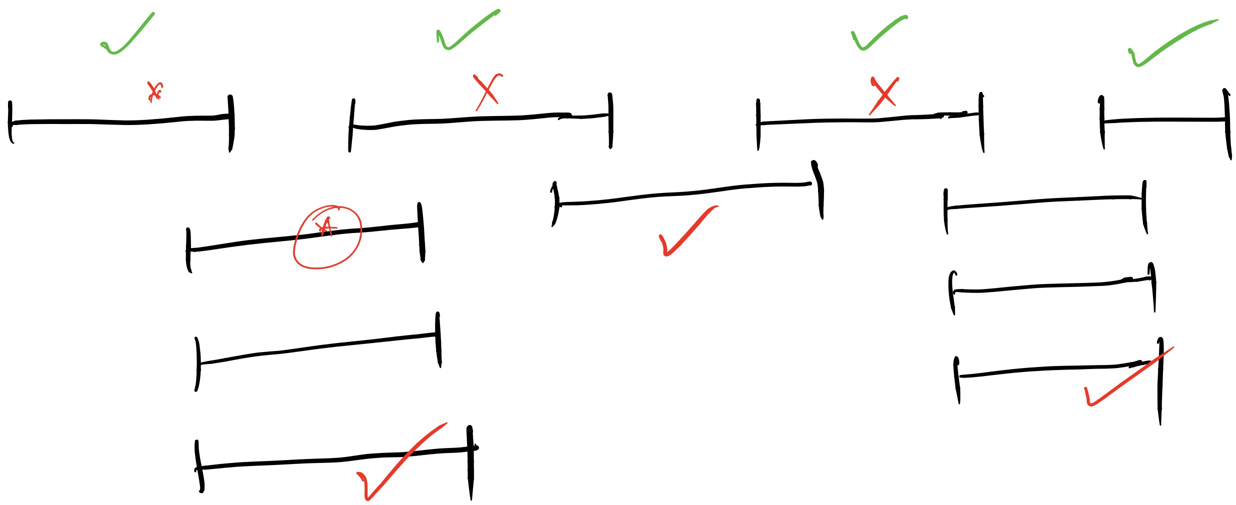
Counterexample:



2 X ~~3~~

Attempt 2: Pick least conflicting job.

Counterexample:

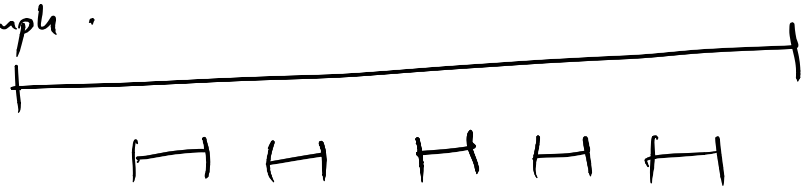


3

Attempt 3: Sort by start time. Start with earliest start time.

(a) Yes (b) No

Counterexample.



Algorithm that works

$|R| = n$

Let R be set of all requests
A: schedule

(i) $\emptyset \leftarrow A$

(ii) while R is not empty

find $i \in R$ with least $f(i)$

Add i to A. Remove conflicting jobs from R

endwhile.

<u>Run time:</u>

(i) Sorting jobs by $f(i)$:

$O(n \log n)$

<u>Proof of correctness</u>

<u>Obs. 1</u> A is a compatible schedule.

<u>Pf:</u> Suppose $i, j \in A$, and they conflict.

i is scheduled before j.
Then j must have been removed from R.
Contradiction.

Strategy: 'Staying ahead of optimal'.

Pf: Let O be an optimal schedule.

$$A = \{i_1, \ldots, i_k\}$$
$$O = \{j_1, \ldots, j_m\}$$

(sorted by finish times)

Claim: $\forall\ l \leq k$,

$$f(i_l) \leq f(j_l).$$

Pf: Use induction (on l).

Base case. $l = 1$. Direct from algorithm.
Inductive step: Assume $f(i_{l-1}) \leq f(j_{l-1})$

$(\ell \geq 2)$

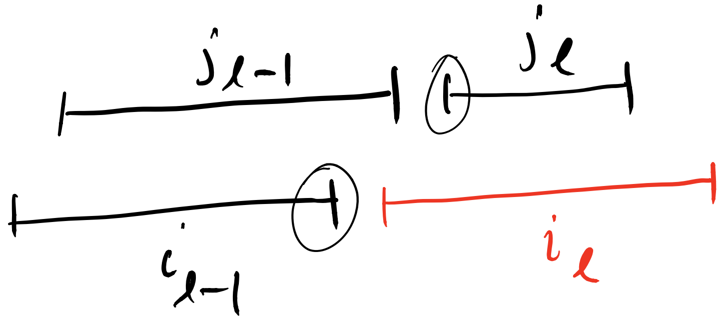
Suppose $f(j_\ell) < f(i_\ell)$

Clearly,

$$s(j_\ell) > f(j_{\ell-1})$$

$\Rightarrow \quad s(j_\ell) > f(j_{\ell-1}) \geq f(i_{\ell-1})$

$\Rightarrow \quad j_\ell$ does NOT conflict with $i_{\ell-1}$.



Contradiction!

Claim: A is a maximal compatible schedule.

Pf: Let $m > k$.

We know, $f(i_k) \leq f(j_k)$

(by claim 1)

Thus, $s(j_{k+1}) > f(j_k) \geq f(i_k)$

$\Rightarrow$ j_{k+1} is compatible with i_k.

Contradiction!

"Proof by certificate"

$|R| = n$

Let R be set of all requests

A: schedule

(i) $\phi \leftarrow A$

(ii) while R is not empty

find $i \in R$ with least $f(i)$

Add i to A. Remove

conflicting jobs from R
endwhile.

$C(i)$

(note that job i conflicts with itself. Thus, $i \in C(i)$)

Claim: $C'(i_1), C'(i_2), \ldots, C'(i_k)$

(where $|A| = k$.)

is a partition of R.

Pf: $\bigcup_{j=1}^{k} C'(i_j) = R$.

each j is exactly in one of $C'(i)$: since each job is exactly removed once.

$\square$.

Claim: Any compatible schedule can have at most 1 job/request from any $c'(i')$.

$\Rightarrow$ A is optimal: Any compatible schedule is of size atmost k;

$|A| = k$.