Interval Partitioning Problem

Input: n lectures / jobs.

each lecture has a start time.

s_j

and finish time

f_j

(for lecture j)

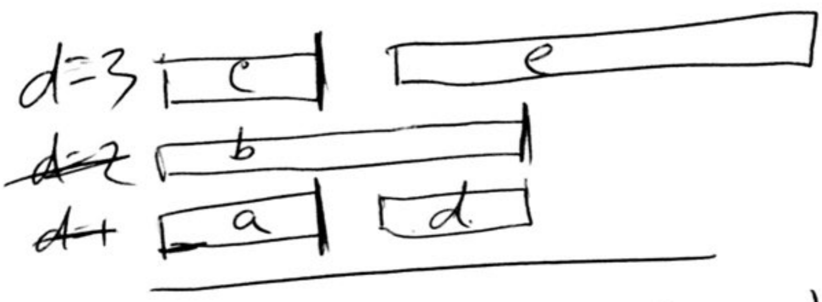
Find: minimum number of classrooms needed to schedule all lectures without conflicts.

Def: The depth of a set of open intervals is the maximum number that any given time point.

Observation: Number of classrooms needed is at ~~the~~ least the depth of the input.

Greedy alg: consider each lecture in order of increasing start time and assign it to any available classroom, and open a new classroom if none are available.

To implement, maintain, for each classroom, the finish of the last job added there.



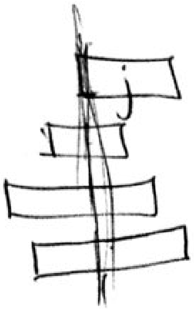
Run time = $O(n \log n)$.

Theorem: Greedy returns a valid schedule (i.e. w/out conflicts) that uses the minimum number of classrooms.

Proof:
- The fact that Greed returns a valid schedule follows directly by the check that the Greedy alg. performs in the first if statement.

- Let d be the # of classrooms allocated by Greedy.
- Classroom d is opened. b/c we needed to schedule a lecture (say, lecture j); that is incompatible

with all the $d-1$ last lectures in other classrooms



- These d lectures all end after s_j
- These d lectures begin no later than s_j (b/c we sorted them, i.e. we are considering in non-decreasing order of start time.
- Thus, we have d lectures overlapping at time $s_j + \epsilon$
- Depth of the input is $\geq d$.
- By our previous observation, we need at least d classrooms.
- This concludes the proof ∎