4 October 2019

Greedy

Given: set of tasks $T = \{[b_i, f_i)\}_{i=1}^{n}$,
find subset of max size s.t.
no 2 tasks overlap

Soln:

Schedule (T)

$\Theta(n \log n)$ Sort T from earliest to latest finish time

$\Theta(1)$ $S \leftarrow \{[b_1, f_1)\}$ // assuming $T \neq \emptyset$
last $\leftarrow f_1$ // finish time of last el. of S.

repeat n times { $\Theta(1)$ for $i = 2 \ldots n$
  if $b_i \geq$ last
    add $[b_i, f_i)$ to S
    last $\leftarrow f_i$
  end if
end for
return S

$\Theta(n \log n) + \Theta(1) + n \cdot \Theta(1) = \Theta(n \log n)$

note: if given sorted array, runtime can be linear b/c we skip the sort step.

problem:

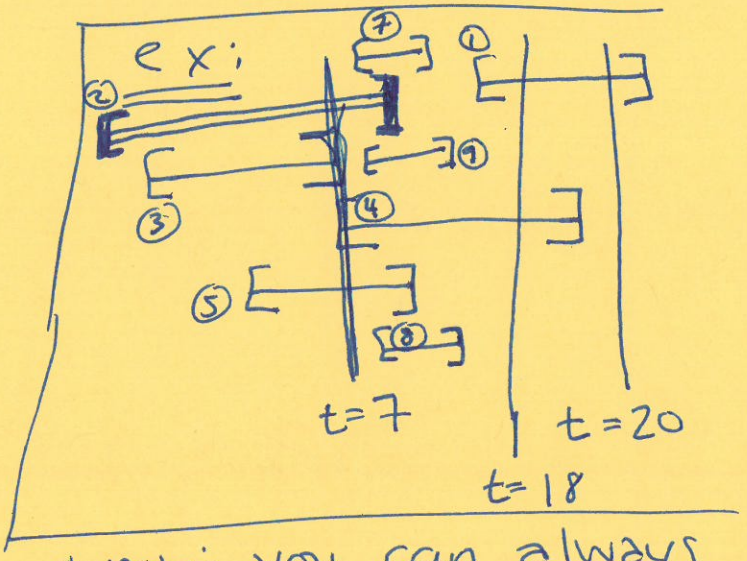
Given $T = \{[b_i, f_i]\}_{i=1}^{n}$

find S such that

① $|S| < \infty$ "S is finite"

② $S \cap t \neq \emptyset \quad \forall t \in T$

③ $|S|$ minimized



Check Tasks (T)

sort by end time

~~times ← [[b1, f1]~~

~~p = 2~~

times ← ∅

while we haven't reached the end of T

add T.pop() (finishtime) to times

pop off additional intervals that overlap that finish time.

end while

return times

nested loop!

- note: you can always choose an endpoint (something to be proven!)
- Consider 1st interval to end.
  - ① no matter what, need a pt in that interval.
  - ② Choosing rightmost endpoint insures max coverage of other intervals.

i = # items that have been popped.

L_i = times is a subset of an optimal solution that covers all popped intervals and does not cover unpopped intervals.

In groups: Section 17.8: largest rectangle

(1) The x-coords of the rectangle must be start/end coords of buildings

(2) Brute Force, try every start/end pair is $\Theta(n^3)$. Can you explain why?

(3) Another approach can be $\Theta(n^2)$.

also, $\Theta(n)$. ← real goal!