Announcements

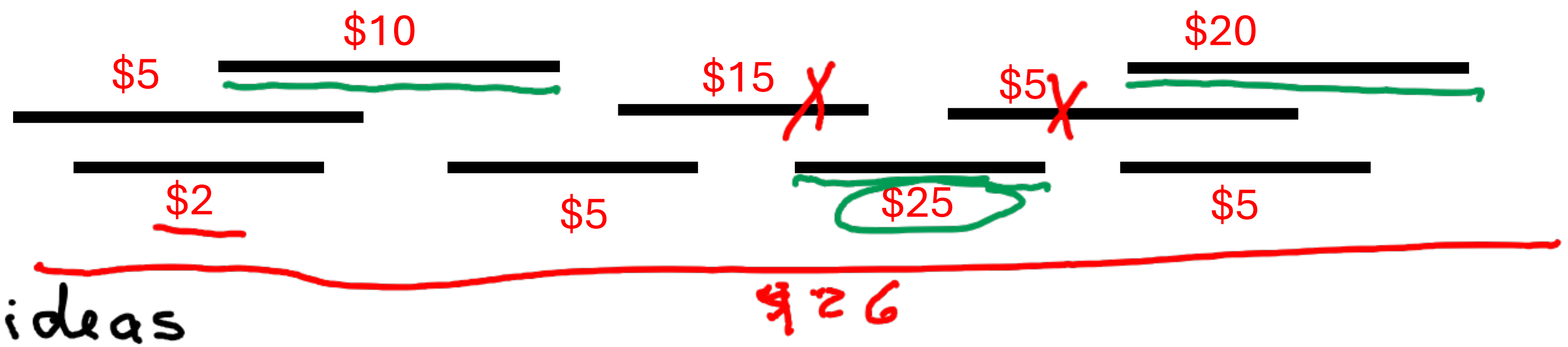
- Short video instead of Monday’s lecture on web
- Hw1 is available on Gradescope. **Due Friday Jan 30**
- On all homework problems (expect for coding) you are **always asked to prove any statement you claim**. If you design an algorithm and claim (1) it is correct and (2) runs in polynomial time, you **must prove both statements**.
- Solutions to section problems posted on canvas, video will be posted shortly due to cancelled sections
- Sections **attendance mandatory, will include a 10 min quiz** about previous hw.

Dynamic Programming I: Weighted Interval scheduling.

Section 6 of KT

The problem: given n intervals $[s(i), f(i)]$ of value v_i, select disjoint intervals of maximum total value

Example:



Greedy ideas

~~1.~~ earliest finish time : ignores values, bad

~~2.~~ highest value first e.g. all value same

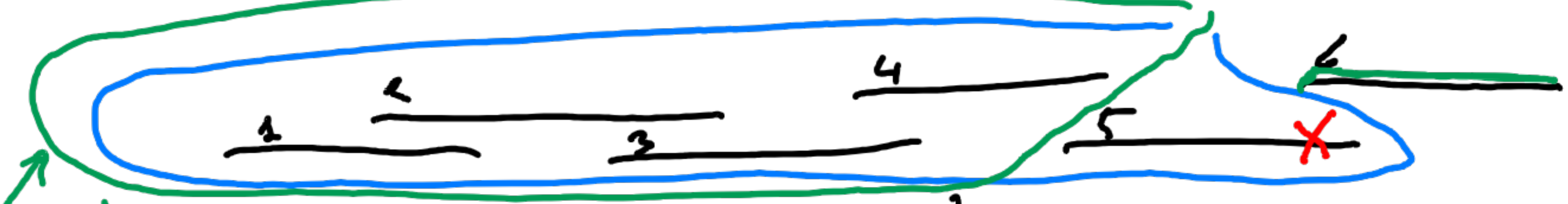
Don't know greedy

Join by Web PollEv.com/evatardos772

- Does greedily adding most valuable interval work?

- Yes/no

Solving problem recursively: the idea



need to delete intervals from end $f(i) \geq s(n)$

$Opt(p(n))$ smaller ~ fewer intervals

suppose: sort by end time $f(i)$

max value $Opt(i)$ for intervals $[1, 2, \dots, i]$

Optimum $Opt(n)$: should n be included

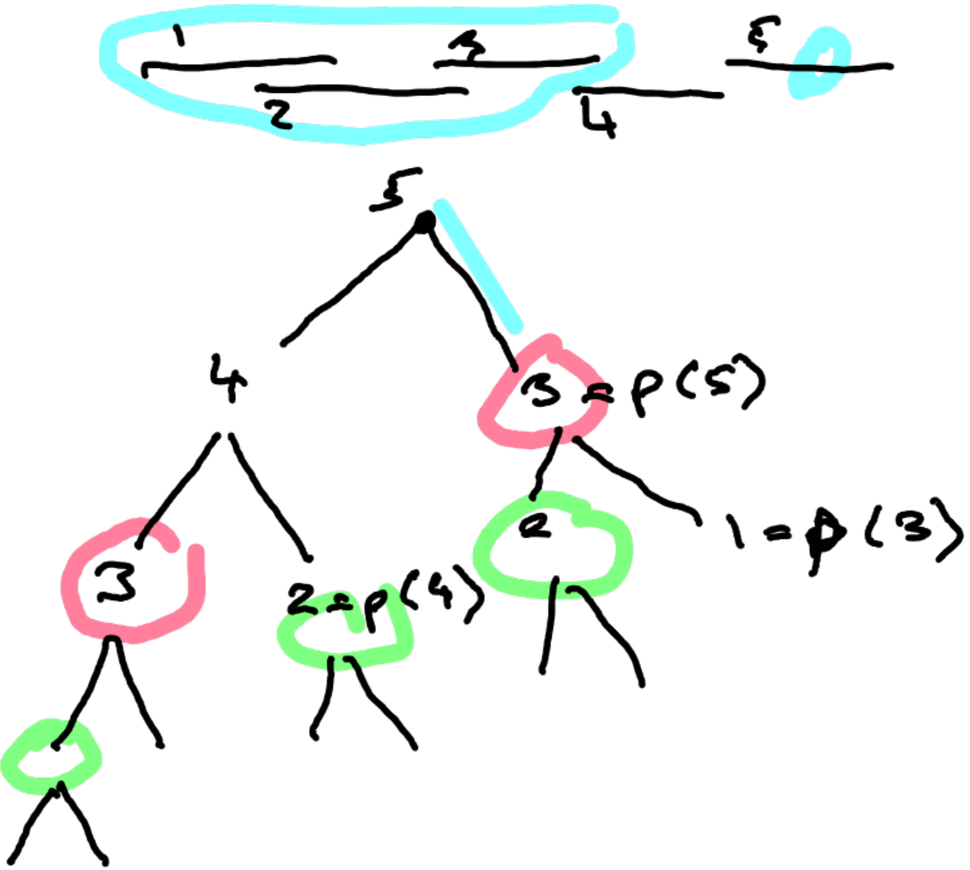
if no: $Opt(n) = Opt(n-1)$

if yes: $Opt(n) = v_n + Opt(p(n))$

$$\boxed{p(i) = \max_j \{ f(j) < s(i) \}}$$

$$Opt(n) = \max\left(Opt(n-1),\ v_n + Opt(p(n))\right)$$

Recursive running time?



Recursive alg

$Opt(0) = 0$, $Opt(1) = v_1$

if $n = 0$ or 1 see above

Else $Opt(n) = \max(Opt(n-1),$
$Opt(p(n)) + v_n)$

trouble: recursive version
run exponential time

keeps repeating computation

Iterative version: memorization

needed

Opt (i)

$Opt(0) = 0$

$Opt(1) = v_1$

For $i = 2, \dots, n$

* $Opt(i) = \max(Opt(i-1), v_i + Opt(p(i)))$

endfor

Running time: sort by $f(i)$ $O(n \log n)$

find $p(i)$ all i $O(\log n)$ each as $f(i)$ sorted

use binary search

$O(n \log n)$

running for loop $O(n)$

Correctness

induction

base: $i = 0, 1$ obvious

induction step

claim $Opt(i)$ correct

assuming ind. hyp $j < i$ $Opt(j)$ correct

argument see above

Extracting the Solution (not only the value)

Opt table above finds Opt value & not solution

$Opt(0) = 0$, $Opt(1) = v_1$

For $i = 2, \dots, n$

$Opt(i) = \max[Opt(i-1),\ v_i + Opt(p(i))]$ ← and $Sol(i)$ = record solution also

endfor

Idea 1: make table include solution ← running time $O(n^2)$

Idea 2: (keep $O(n \log n)$)

is n included in solution: yes if max was $v_n + Opt(p(n))$

going backwards on table extracting solution $O(n)$ time