Lecture 30

CSE 331

Nov 11, 2024

Final exam conflict

note @274 stop following **0 views**

Actions

Final exam conflicts

I know some of you have an exam conflict with CSE 331 final exam. Since I'm not sure if I know the exact set of students with conflict, I figured I'll do a piazza post.

If you have an exam conflict with the CSE 331 final please EMAIL me by 5pm on Friday, Nov 15. If you email me after this deadline, I cannot promise to be able to give you a makeup option that works with your schedule.

Please note that the makeup final will be on *Monday, Dec 16* (i.e. a day before the scheduled final exam). My goal is to pick a time that works for everyone on Dec 16.

So *if you email me for a makeup final exam*, please send me all the time(s) that you do a makeup on Monday, Dec 16 between 9am-5pm.

final

Edit good note | 0

Updated 41 seconds ago by Atri Rudra

Reflections 2+1 grading timeline

note @281

stop following **11 views**

Actions

Timeline on project reflections grading

As a heads up, I'll be manually grading all the reflections 1+2 grading so it'll take a bit of time. Since reflection 2 in structure is closer to reflections 3-5, I'll grade reflection 2 first and then reflection 1. My hope is to get reflection 2 graded within 2 weeks.

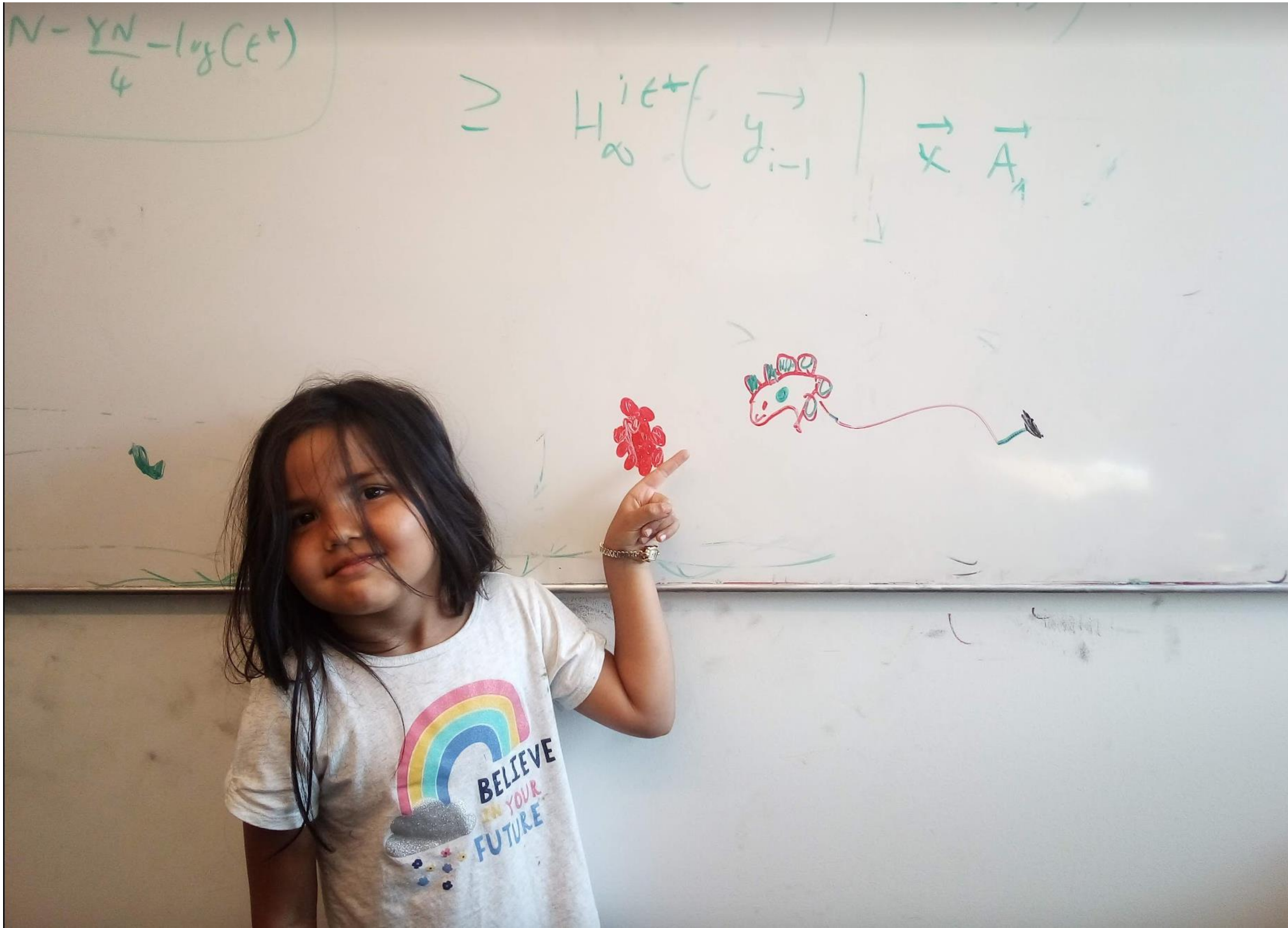
Figured should y'all a heads up as y'all work on the rest of the project.

project

Edit good note | 0

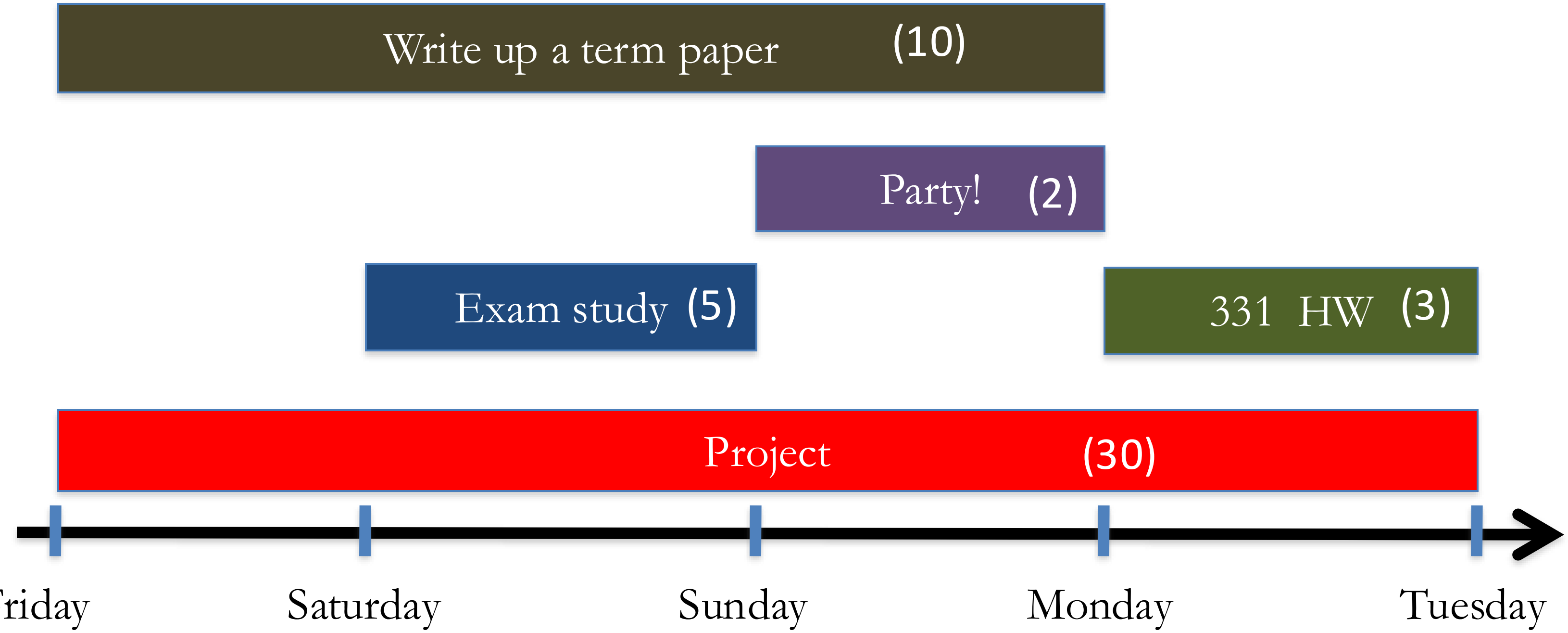
Updated 8 minutes ago by Atri Rudra

Questions/Comments?



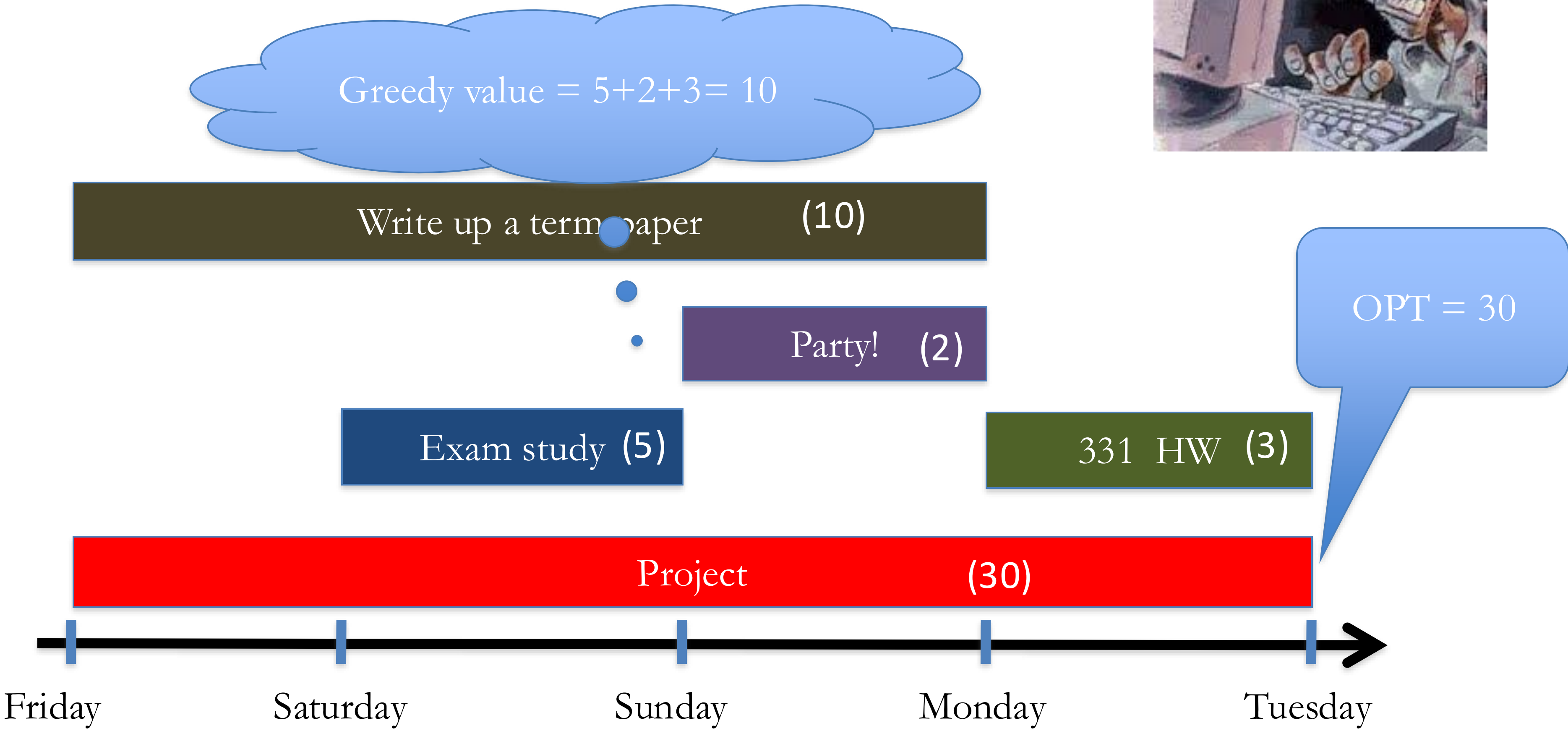
End of Semester blues

Can only do one thing at any day: what is the optimal schedule to obtain maximum value?



Previous Greedy algorithm

Order by end time and pick jobs greedily



Weighted Interval Scheduling

Input: n jobs (s_i, f_i, v_i)

Output: A schedule S s.t. no two jobs in S have a conflict

Goal: max $\Sigma_{i \text{ in } S} v_j$

Assume: jobs are sorted by their finish time

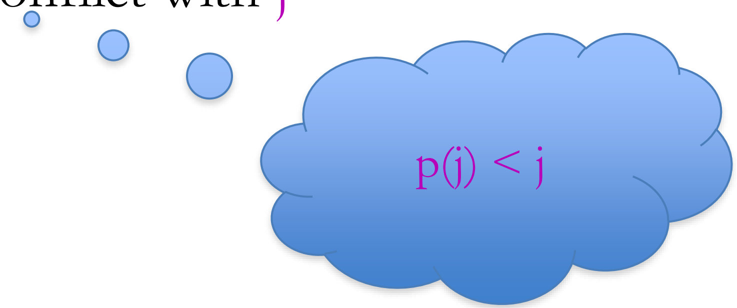
Today's agenda

Finish designing a recursive algorithm for the problem

Couple more definitions

$p(j)$ = largest $i < j$ s.t. i does not conflict with j

= 0 if no such i exists



$OPT(j)$ = optimal value on instance $1,..,j$

Moving to the board…

Property of OPT

j in OPT(j)

j not in OPT(j)

$$OPT(j) = \max \{ v_j + OPT(p(j)),\ OPT(j-1) \}$$

Given OPT(1),…., OPT(j-1), how can one figure out if j in optimal solution or not?


?
ICANHASCHEEZBURGER.COM

A recursive algorithm

Correct for $j=0$

Proof of correctness by induction on j

Compute-Opt(j)

If $j = 0$ then return 0

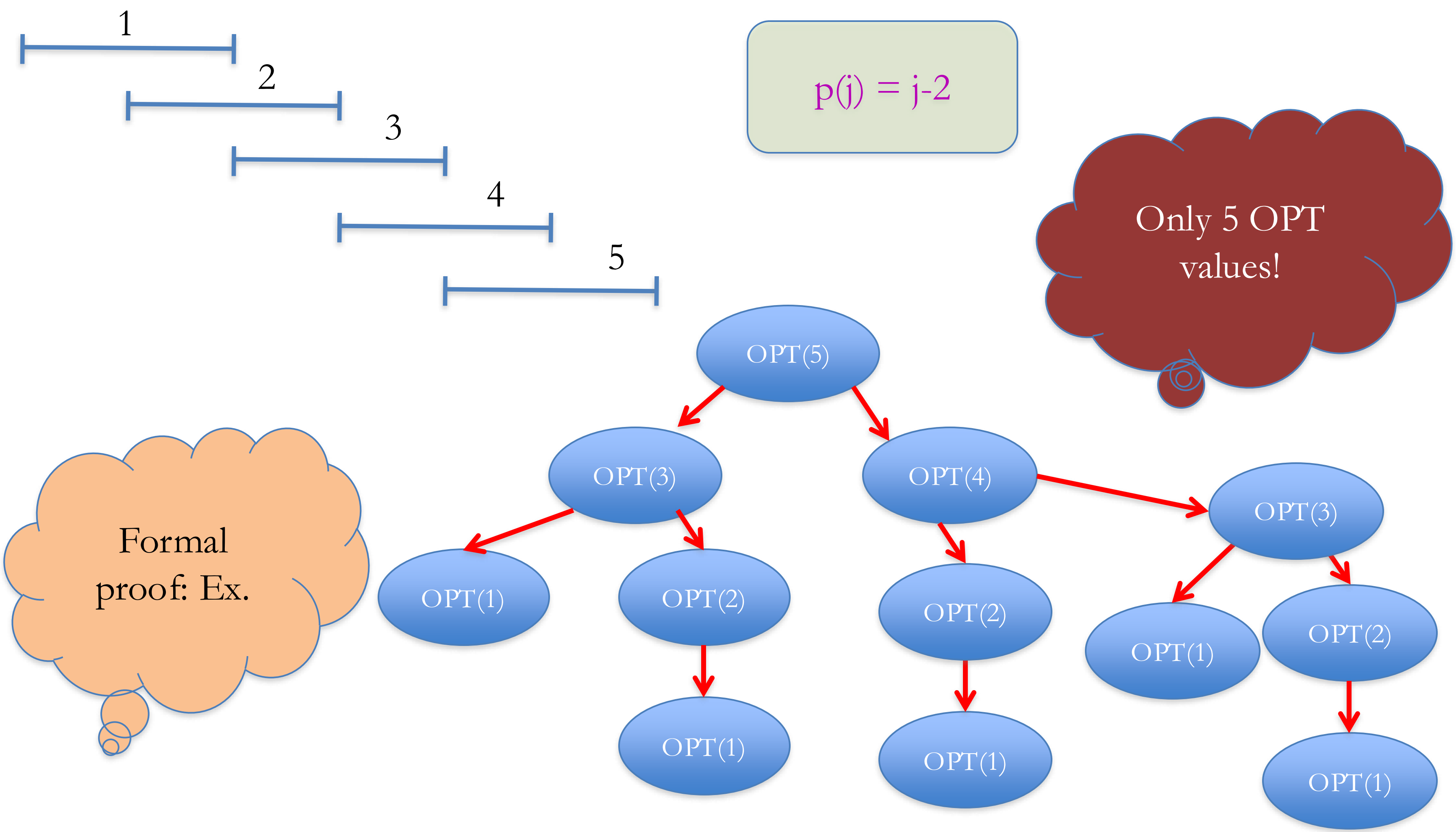
return max { v_j + Compute-Opt($p(j)$), Compute-Opt($j-1$) }

$= OPT(\,p(j)\,)$

$= OPT(\,j-1\,)$

$$OPT(j) = \max \{ v_j + OPT(\,p(j)\,), OPT(j-1) \}$$

Exponential Running Time




?
ICANHASCHEEZBURGER.COM

Using Memory to be smarter

Using more space can reduce runtime!

How many distinct OPT values?

A recursive algorithm

M-Compute-Opt(j)

M-Compute-Opt(j) = OPT(j)

If j = 0 then return 0

If M[j] is not null then return M[j]

M[j] = max { v_j + M-Compute-Opt(p(j)), M-Compute-Opt(j-1) }

return M[j]

Run time = O(# recursive calls)