Lecture 29

CSE 331

Nov 10, 2021

Please have a face mask on

Masking requirement

UR requires all students, employees and visitors – regardless of their vaccination status – to wear face coverings while inside campus buildings.

https://www.buffalo.edu/coronavirus/health-and-safety/health-safety-guidelines.html

Homework 6 out

Homework 6

- Part **(b)**: Present a divide and conquer algorithm that given non-negative integers a and n computes `Power` (a, n) in $O(\log n)$ time.

> **Important Note**
> **To get credit you must present a recursive divide and conquer algorithm and then analyze its running time by solving a recurrence relation. If you present an algorithm that is not a divide and conquer algorithm you will get a level 0 on this entire part.**

Question 1 (Exponentiation) [50 points]

The Problem

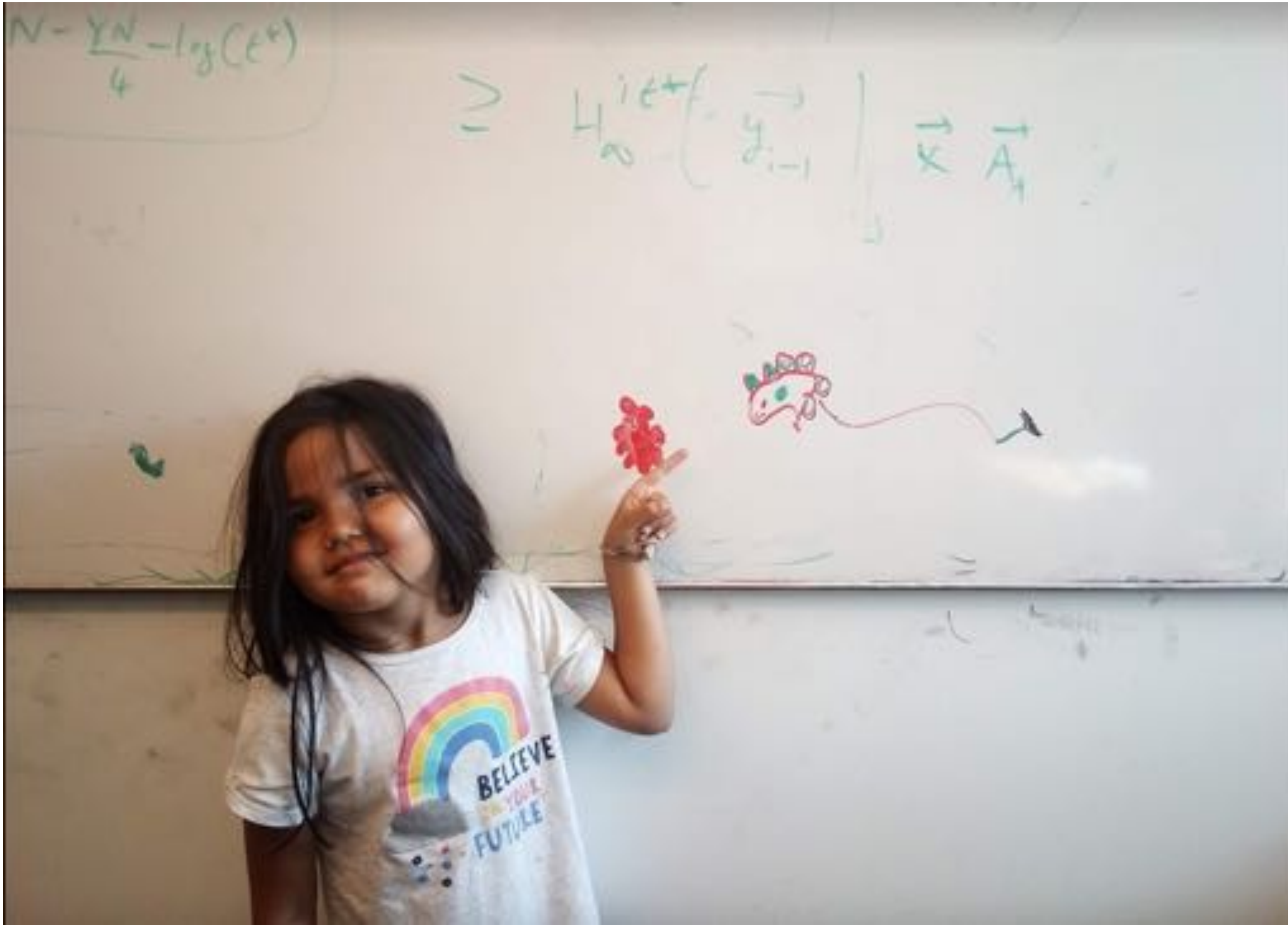
We will consider the problem of exponentiating an integer to another. In particular, for non-negative integers a and n, define `Power` (a, n) be the number a^n. (For this problem assume that you can multiply two integers in $O(1)$ time.) Here are the two parts of the problem:

- Part **(a)**: Present a naive algorithm that given non-negative integers a and n computes `Power` (a, n) in time $O(n)$.

> **Note**
> For this part, there is no need to prove correctness of the naive algorithm but you do need a runtime analysis.

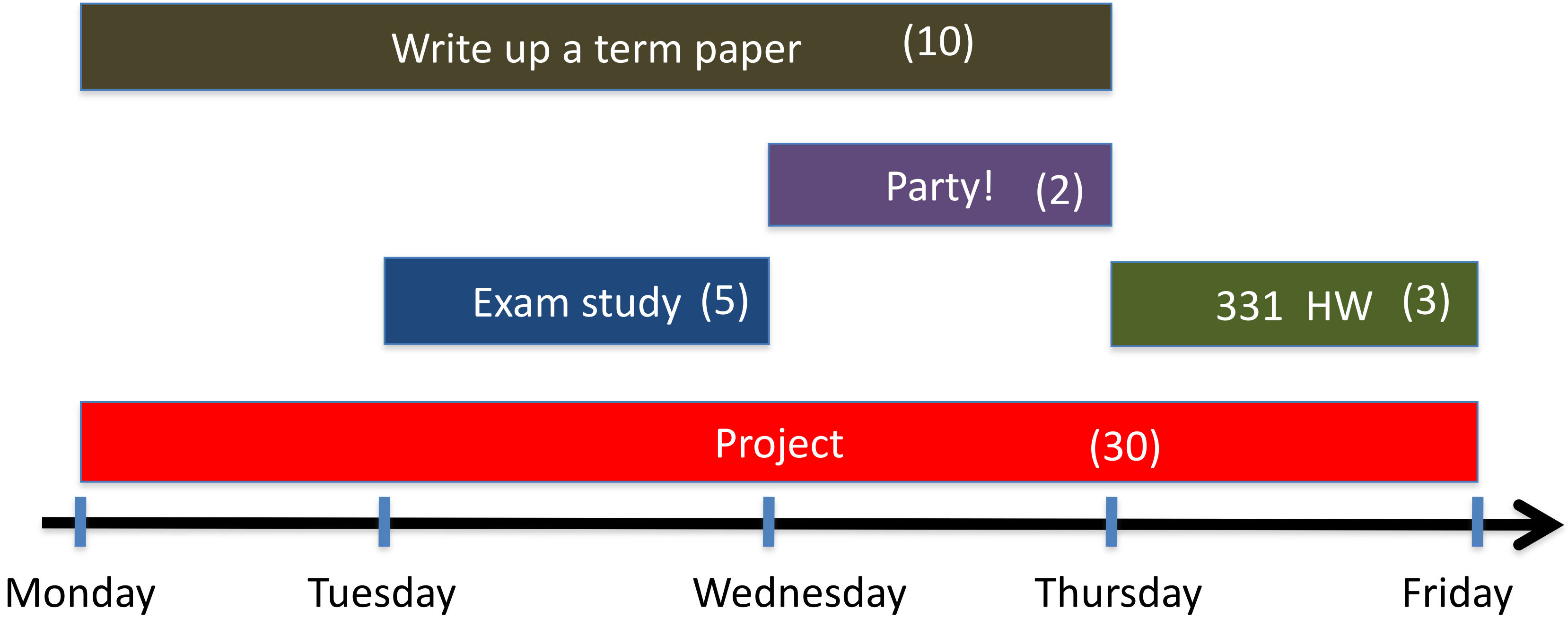
- Part **(b)**: Present a divide and conquer algorithm that given non-negative integers a and n computes `Power` (a, n) in $O(\log n)$ time.

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

Greedy value = 5+2+3= 10

Write up a term paper (10)

Party! (2)

Exam study (5)

331 HW (3)

OPT = 30

Project (30)

Monday Tuesday Wednesday Thursday Friday

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

$p(j) < j$

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

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?

A recursive algorithm

Compute-Opt(j)

Correct for j=0

Proof of correctness by induction on j

If $j = 0$ then return 0

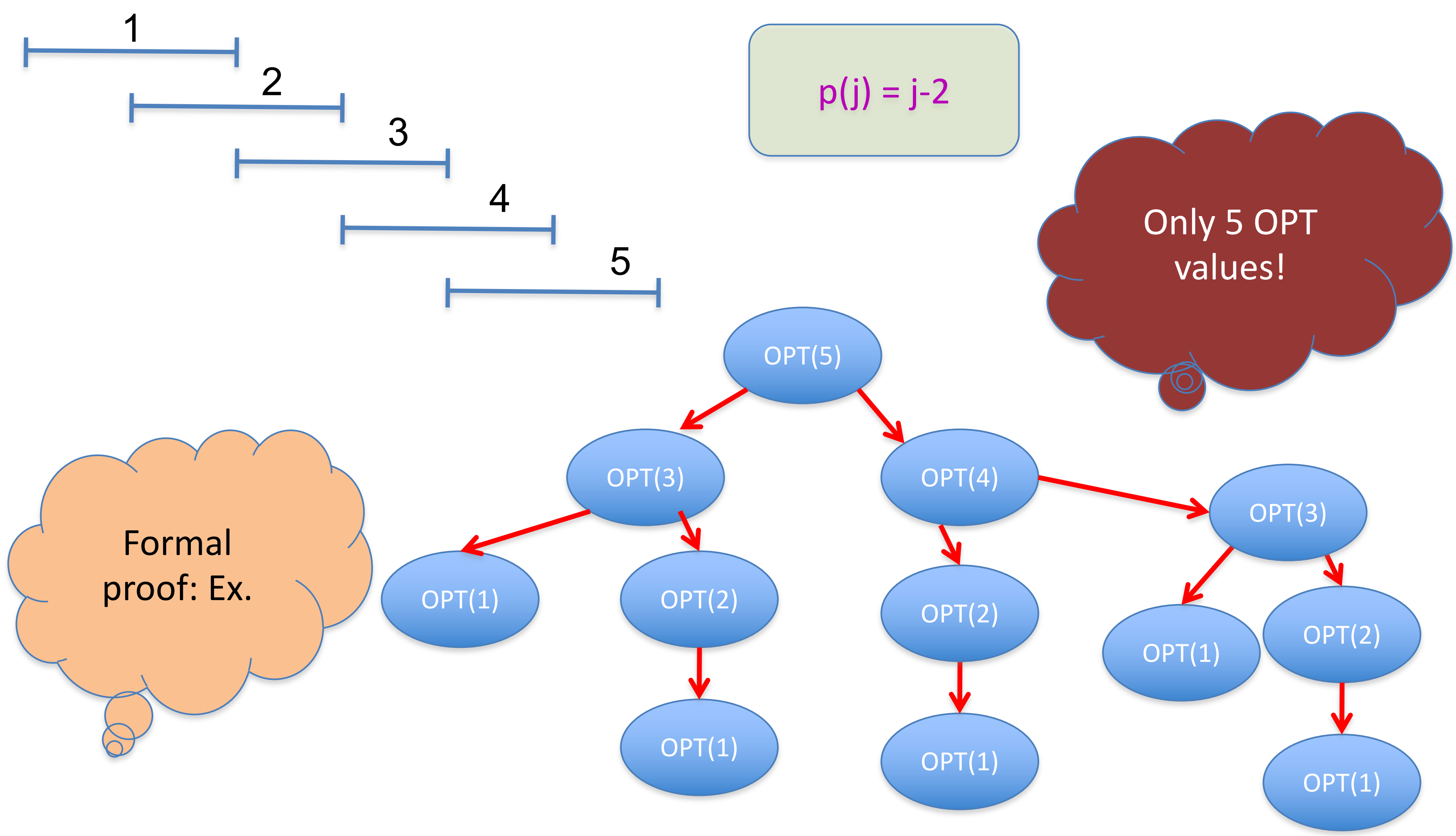
return max { v_j + Compute-Opt(p(j)), Compute-Opt(j-1) }

= OPT(p(j))

= OPT(j-1)

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

Exponential Running Time



Using Memory to be smarter

Using more space can reduce runtime!

How many distinct OPT values?

A recursive algorithm

M-Compute-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]

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

Run time = O(# recursive calls)