Lecture 18

CSE 331

Oct 11, 2024

Grading update

Mid terms should hopefully be handed back next Th

Interval Scheduling Problem

Input: n intervals $[s(i), f(i))$ for $1 \leq i \leq n$

Output: A *schedule* S of the n intervals

No two intervals in S conflict

$|S|$ is maximized

Analyzing the algorithm

R: set of requests

Set S to be the empty set

While R is not empty

Choose i in R with the earliest finish time

Add i to S

Remove all requests that conflict with i from R

Return S* = S

S* has no conflicts

S* is an optimal solution

Greedy "stays ahead"

Greedy stays ahead lemma

$S^* = \{i_1,..,i_k\}$

$O = \{j_1,..,j_m\}$

Lemma 1: For all $1 \leq \ell \leq k$

$$f(i_\ell) \leq f(j_\ell)$$

Questions?

Runtime analysis of Greedy Algo.

R: set of requests

Set S to be the empty set — O(1)

While R is not empty — Repeated at most n times

  Choose i in R with the earliest finish time — O(n)

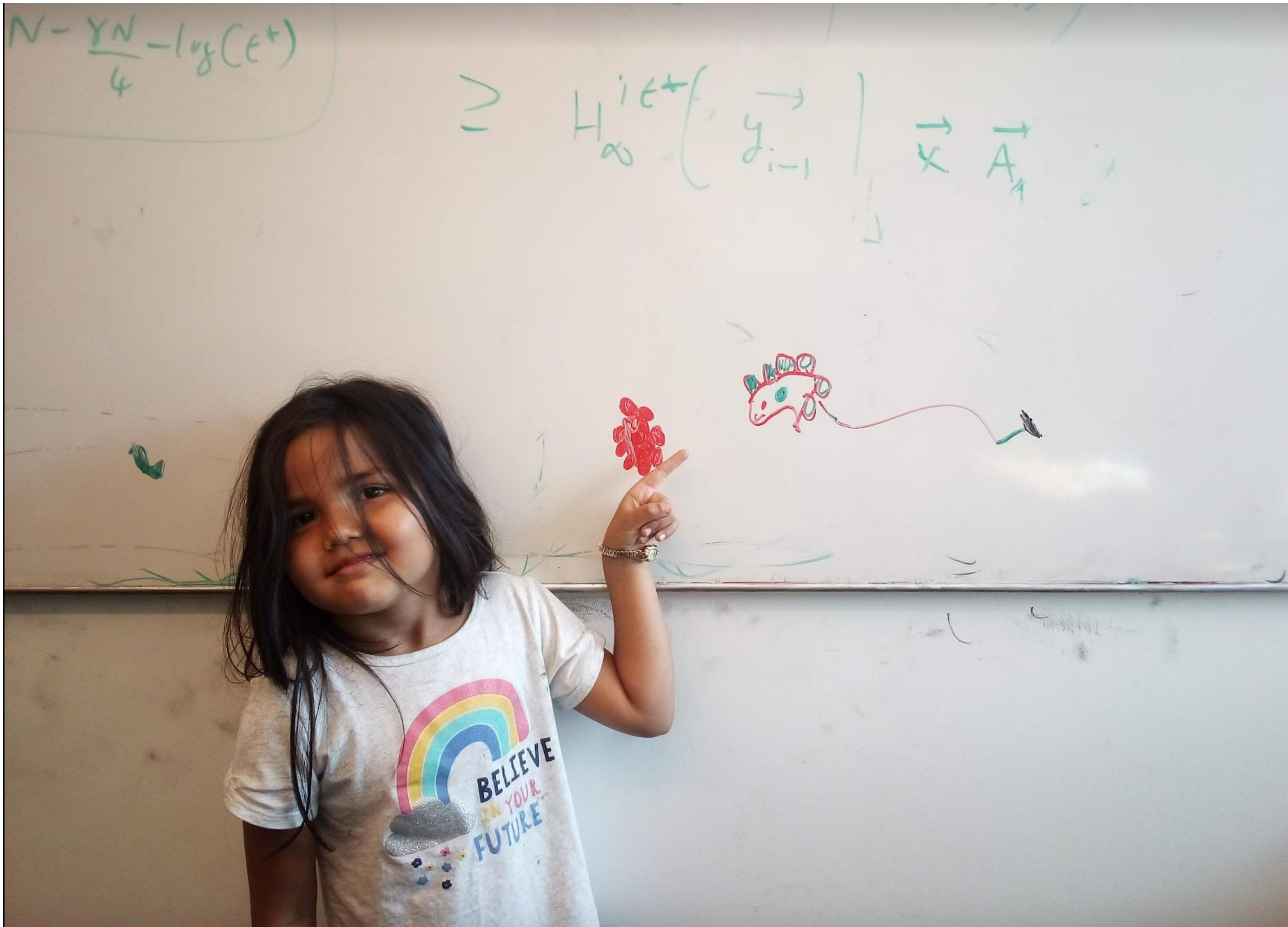
  Add i to S — O(1)

  Remove all requests that conflict with i from R — O(n)

Return S* = S — O(n)

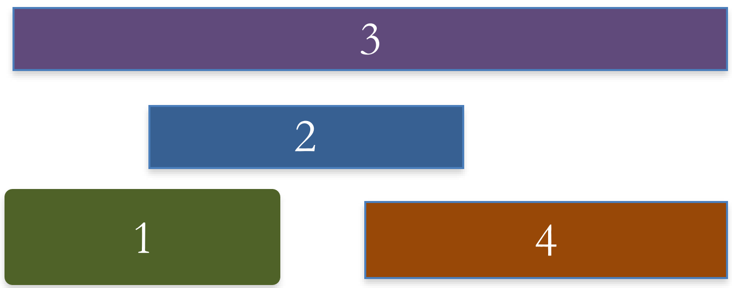
Overall: O(n) + n*O(n) = O(n^2)

Questions/Comments?



Algorithm implementation

Go through the intervals in order of their finish time



Check if s[i] < f(1)

In general, if jth interval is the last one chosen

Pick smallest i>j such that s[i] ≥ f(j)

O(n log n) run time

The final algo

$O(n \log n)$ time sort intervals such that $f(i) \leq f(i+1)$

$O(n)$ time build array $s[1..n]$ s.t. $s[i]$ = start time for i

Add 1 to S and set $f = f(1)$

For $i = 2 .. n$

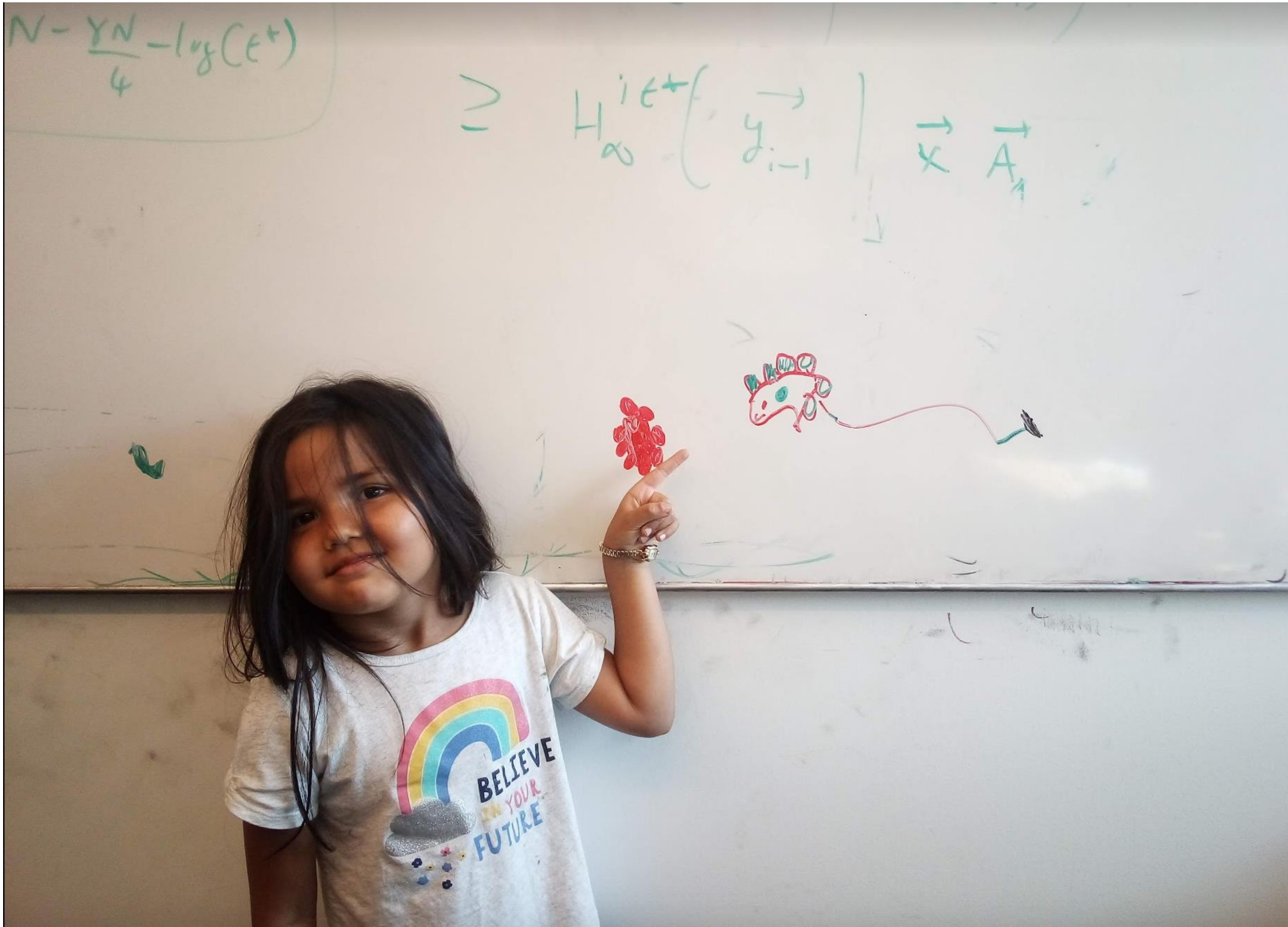
If $s[i] \geq f$

Add i to S

Set $f = f(i)$

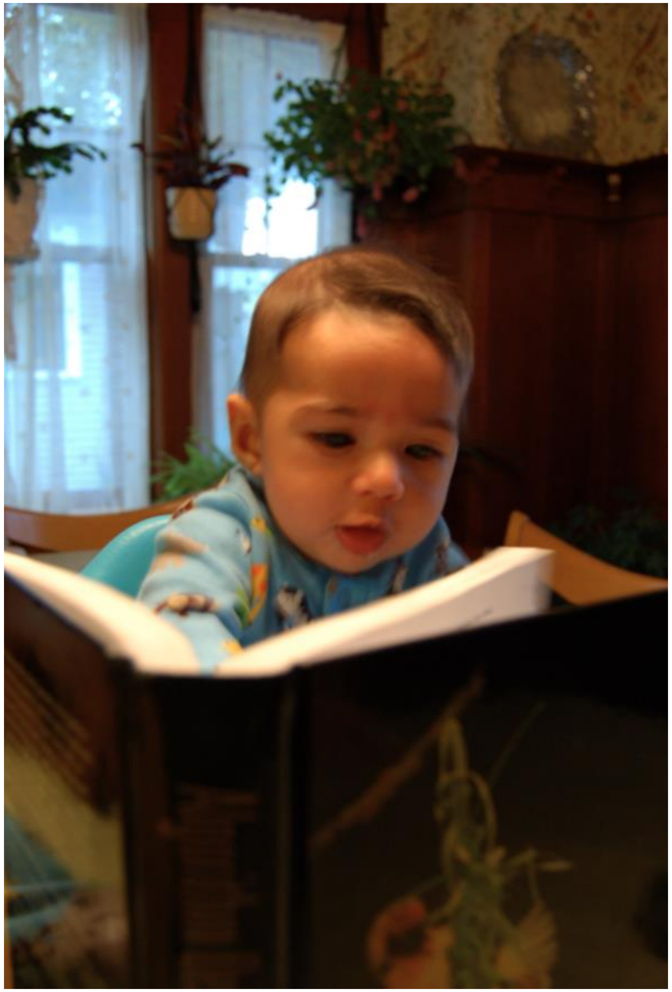
Return $S^* = S$

Questions/Comments?



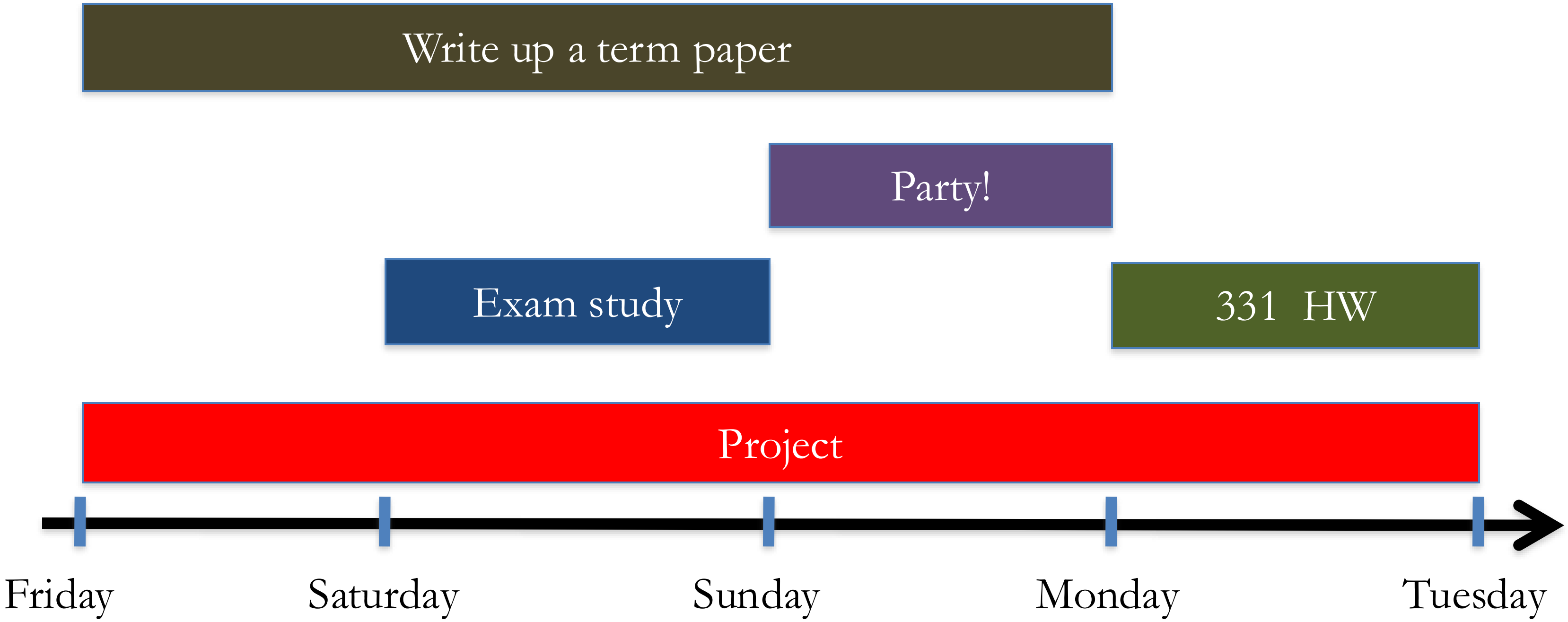
Reading Assignment

Sec 4.1 of [KT]



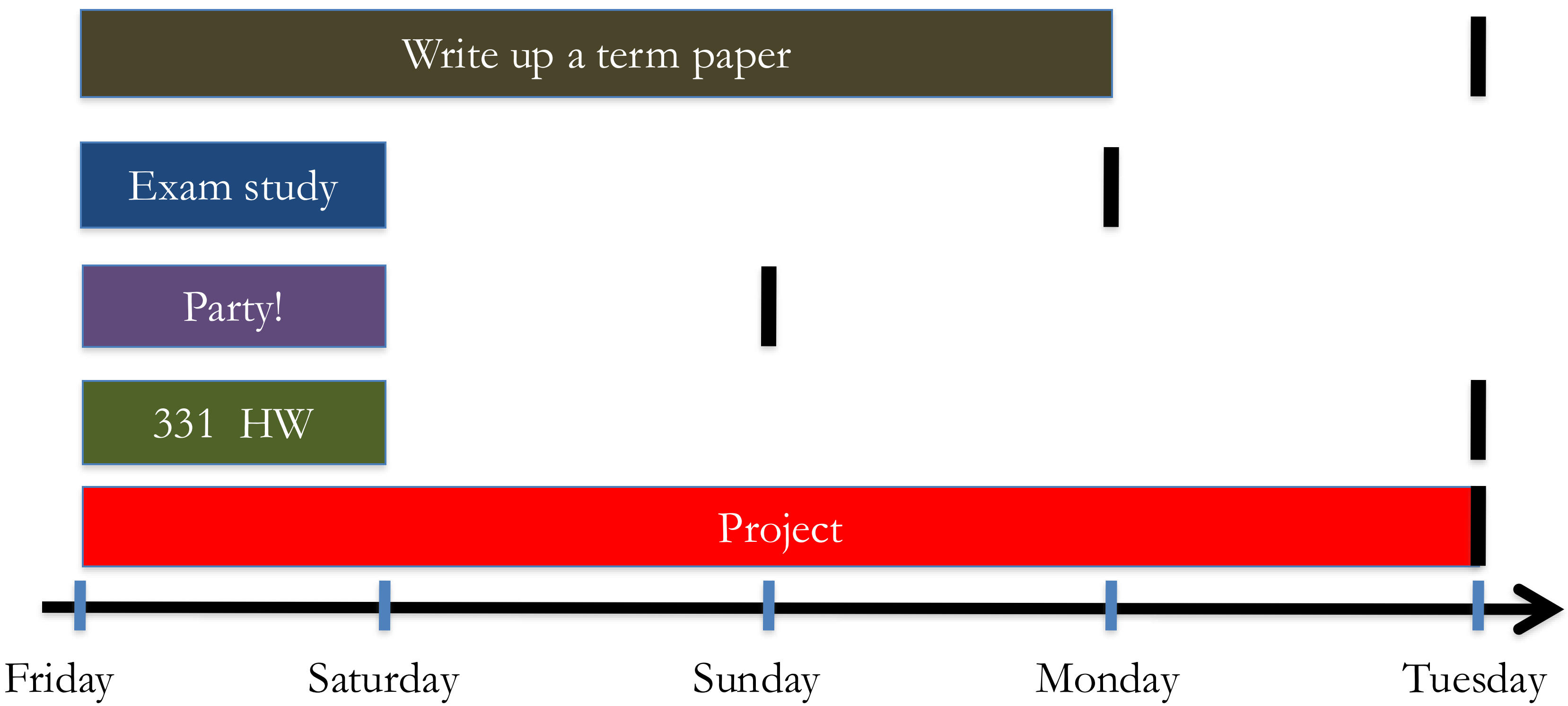
The "real" end of Semester blues

There are deadlines and durations of tasks



The “real” end of Semester blues

There are deadlines and durations of tasks



The algorithmic task

YOU decide when to start each task

Write up a term paper

Exam study

Party!

331 HW

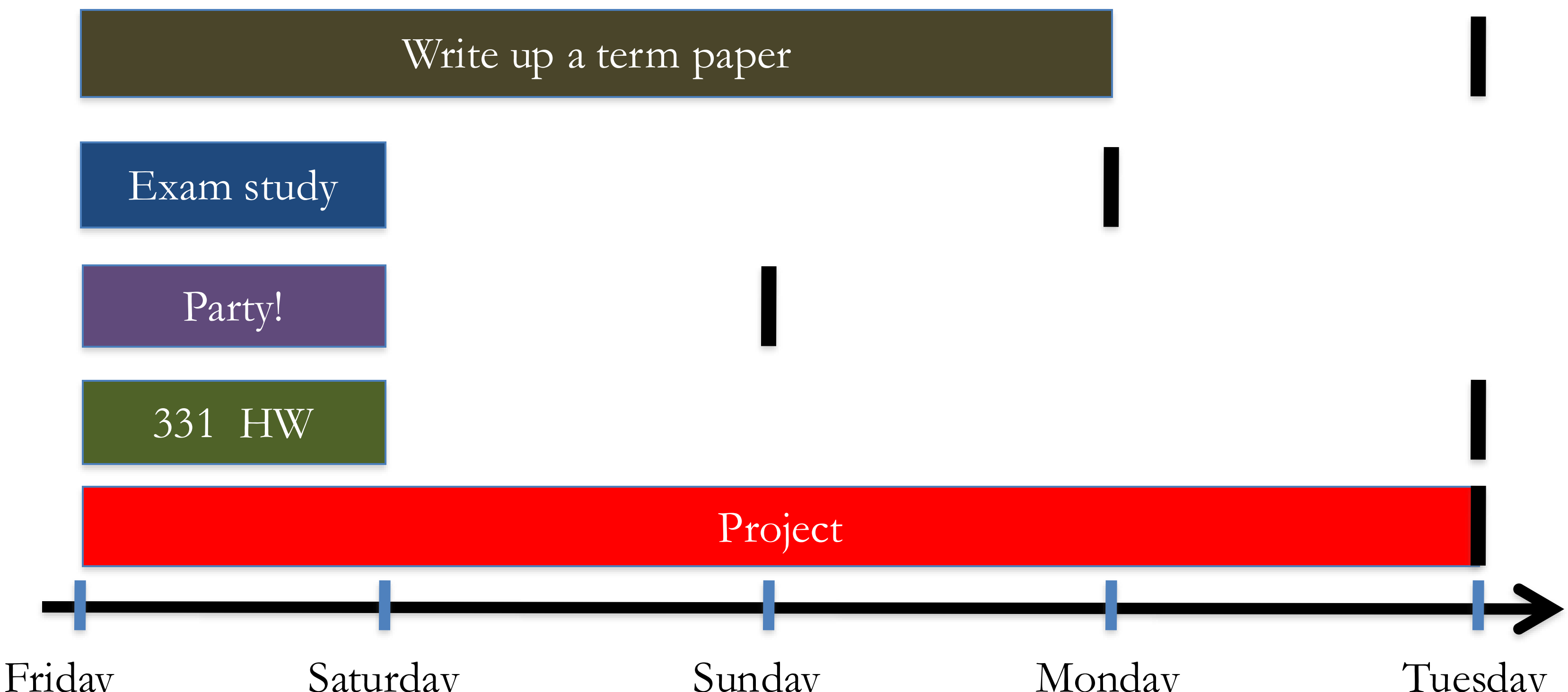
Project

You have to do ALL the tasks

Friday Saturday Sunday Monday Tuesday

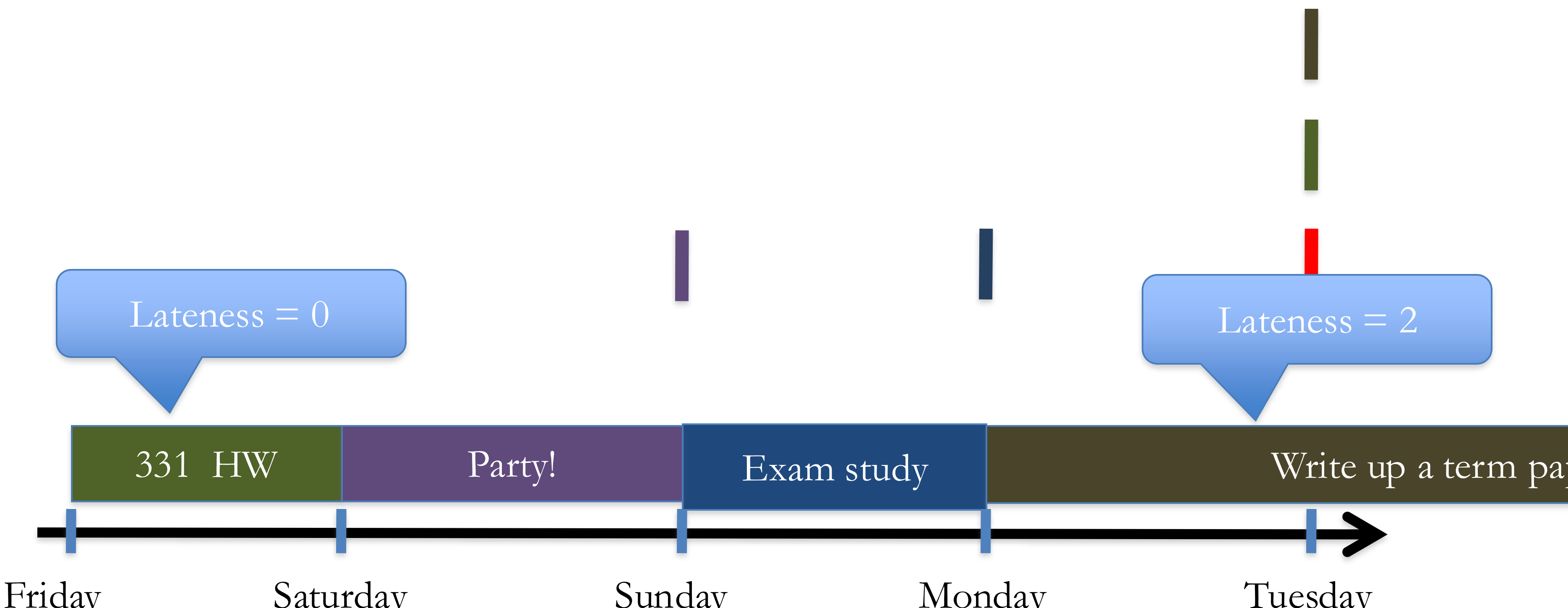
Scheduling to minimize lateness

All the tasks have to be scheduled
GOAL: minimize maximum lateness



One possible schedule

All the tasks have to be scheduled
GOAL: minimize maximum lateness



Minimizing Max Lateness

Minimizing Maximum Lateness

This page collects material from previous incarnations of CSE 331 on scheduling to minimize maximum lateness.

Where does the textbook talk about this?

`Section 4.2` in the textbook has the lowdown on the problem of scheduling to minimize maximum lateness.

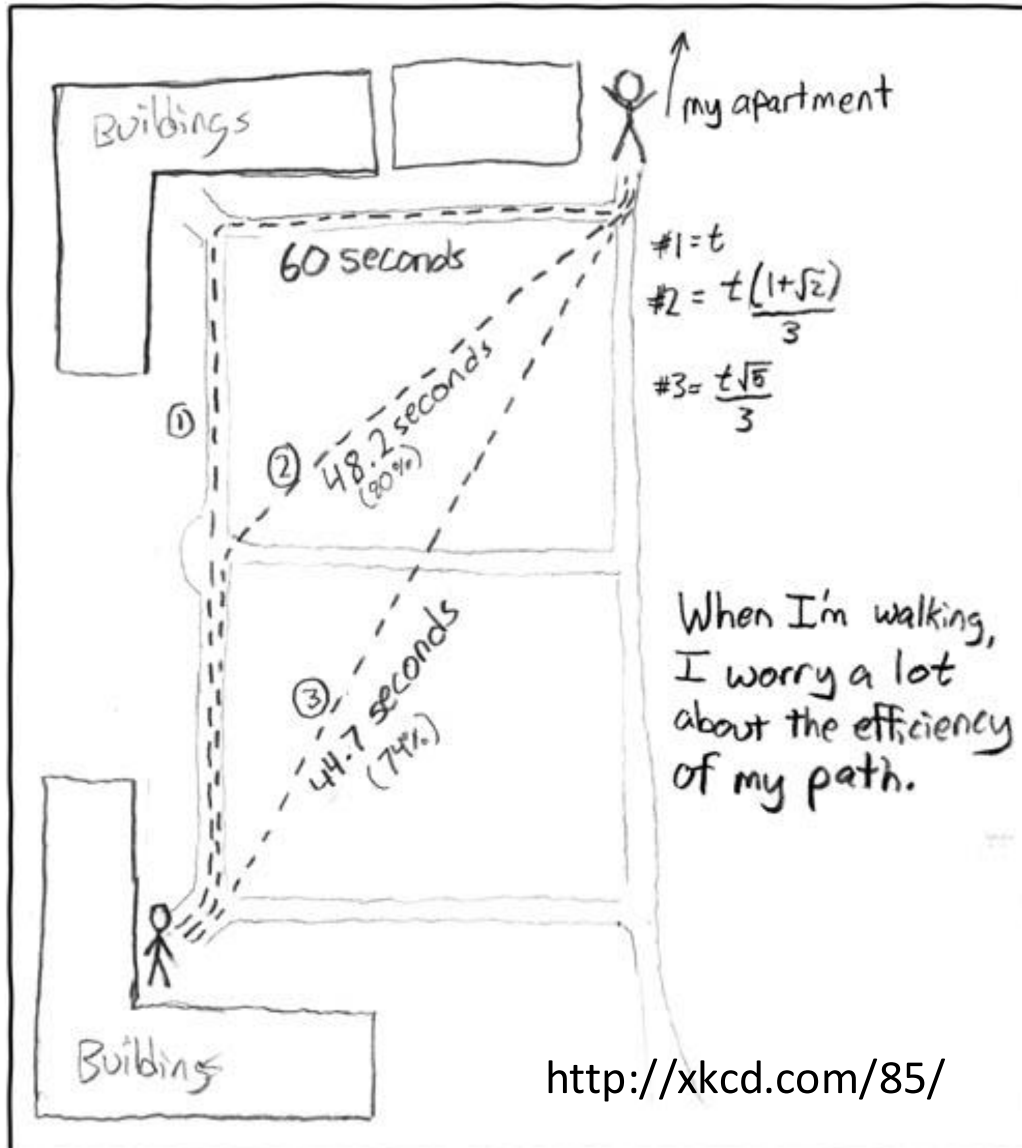
Fall 2018 material

First lecture

Here is the lecture video:

Rest of today

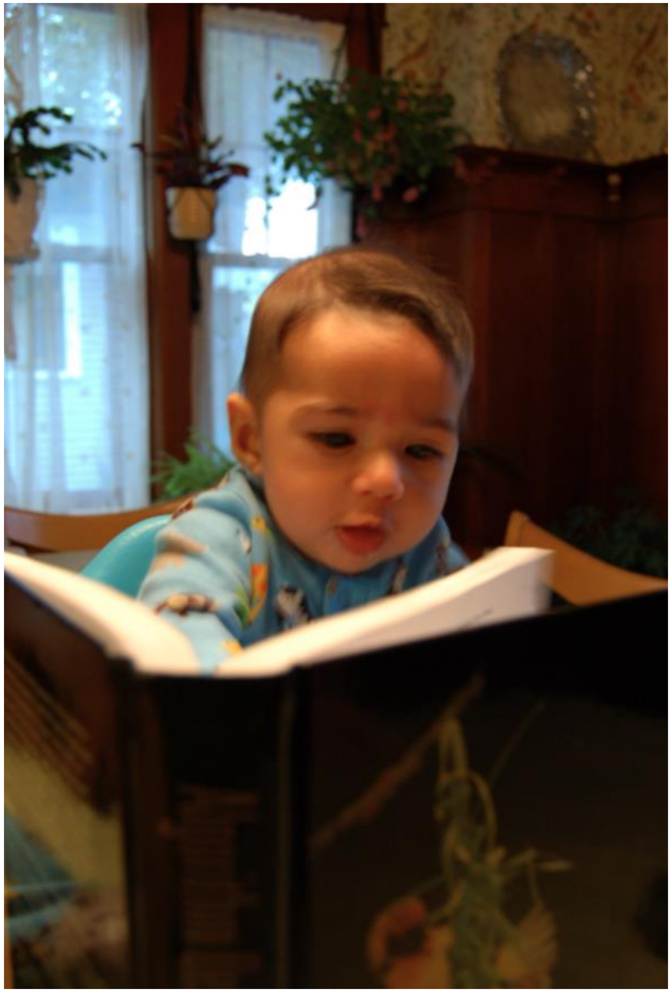
Shortest Path Problem



http://xkcd.com/85/

Reading Assignment

Sec 2.5 of [KT]

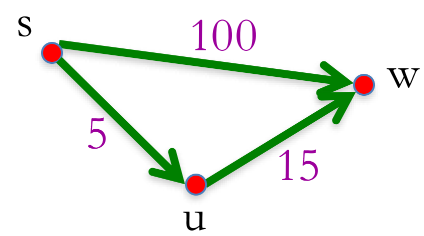


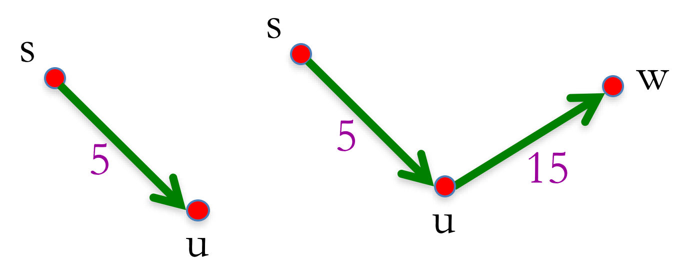
Shortest Path problem

Input: *Directed* graph $G=(V,E)$

Edge lengths, ℓ_e for e in E

"start" vertex s in V





Output: All shortest paths from s to all nodes in V

Naïve Algorithm

$\Omega(n!)$ time

Dijkstra's shortest path algorithm

On to the board…