Lecture 18

CSE 331

Oct 11, 2023

Please do fill in the feedback

note @281

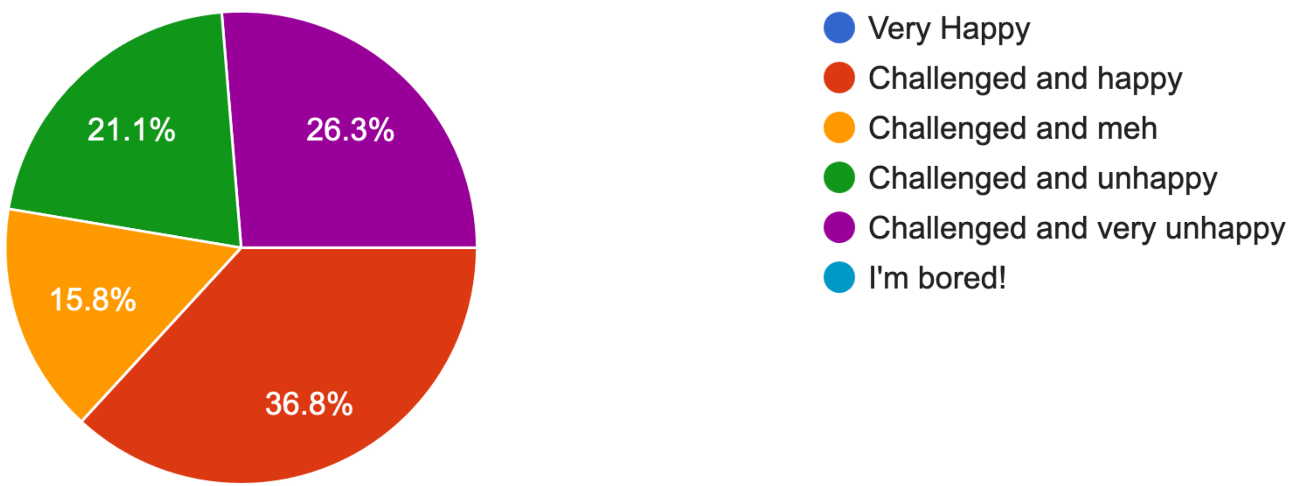
stop following 2 views

Actions

Feedback on CSE 331

Overall your feeling about CSE 331

19 responses



Mid-terms next week

Mid-term 1: next Wed in class

Mid-term 2: Fri next week in class

Grading status

Hopefully by tonight: Quiz 1, HW 3

Project Coding submissions

note @305

stop following 39 views

Actions

Issue with project coding submissions

So it seems like with the latest update to Autolab has broken a part of the grading of the coding projects (currently the issue has been reported for Problem 1 but I suspect the issue will be there for other problems as well).

Specifically, Autolab assigns the correct grade to the student who submits for the group but incorrectly throws an error for the other students in the group.

So going forward:

- I'll be implementing a fix for that and hope to be done **by the end of the week**
  - Once the fix is in place, I'll ask everyone who has already submitted to re-submit.
- In the meantime, y'all can still submit your code to check if your submission works-- it's just that for now please only look at the score assigned to the student who submit.

Thanks to those bought this to my notice and apologies for any inconvenience this might cause y'all.

autolab project

Edit good note | 0

Updated 6 hours ago by Atri Rudra

HW 4 is out

Homework 4

Due by **11:30pm, Tuesday, October 17, 2023**.

Make sure you follow all the homework policies.

All submissions should be done via Autolab.

The care package on minimizing the maximum lateness problem would be useful for Q3 and *might* be useful for Q2(b) as well.

Probably the easiest HW

Question 1 (High Speed Internet) [50 poin

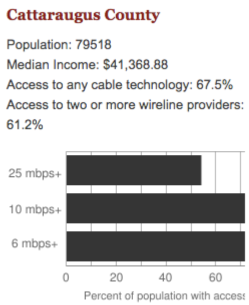
The Problem

We come back to the issue of many USA regions not having high speed internet. In this question, you will consider an algorithr oblem you would ed to solve to help out a (fictional) place get high speed Internet.

You are the algorithms whiz in the effort to bring high speed Internet to `SomePlaceInUSA`. After lots of rounds of discussions and public feedback, it was decided that the most cost-effective way to bring high speed internet to `SomePlaceInUSA` was to install high speed cell towers to connect all houses in `SomePlaceInUSA` to high speed internet. There are two things in your favor:

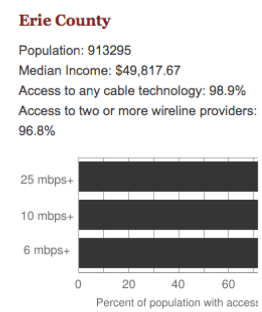
1. It just so happens that all of the n houses in `SomePlaceInUSA` are on the side of a straight road that runs through the town.

Make broadband more available



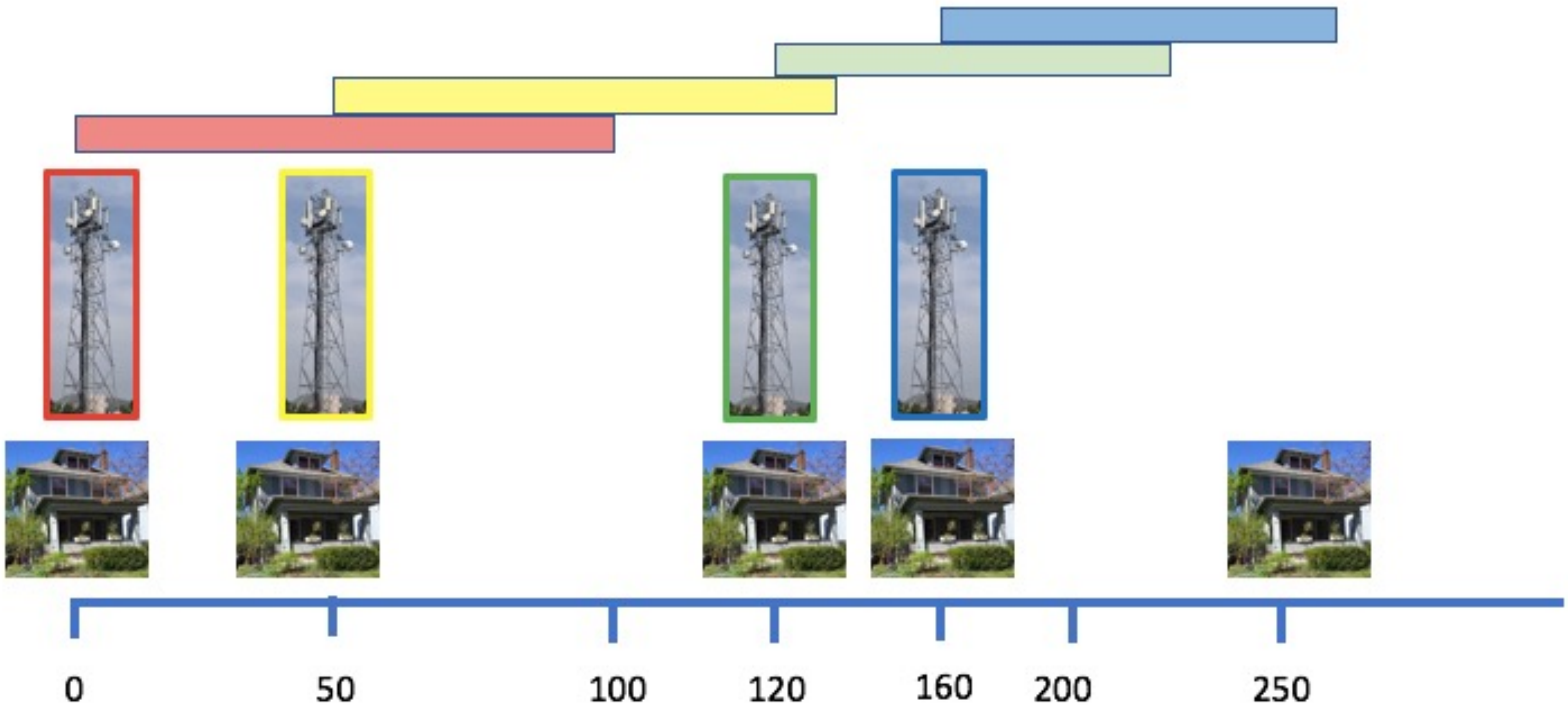
BOTH
technical and
societal issues

Say you are
tasked to come
up with the
infrastructure



HW 4 Q1: How to lay down towers

Here is a quick visual argument for the above leads to continuous cell coverage:



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?

Proof of Lemma 1 on the board…

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)

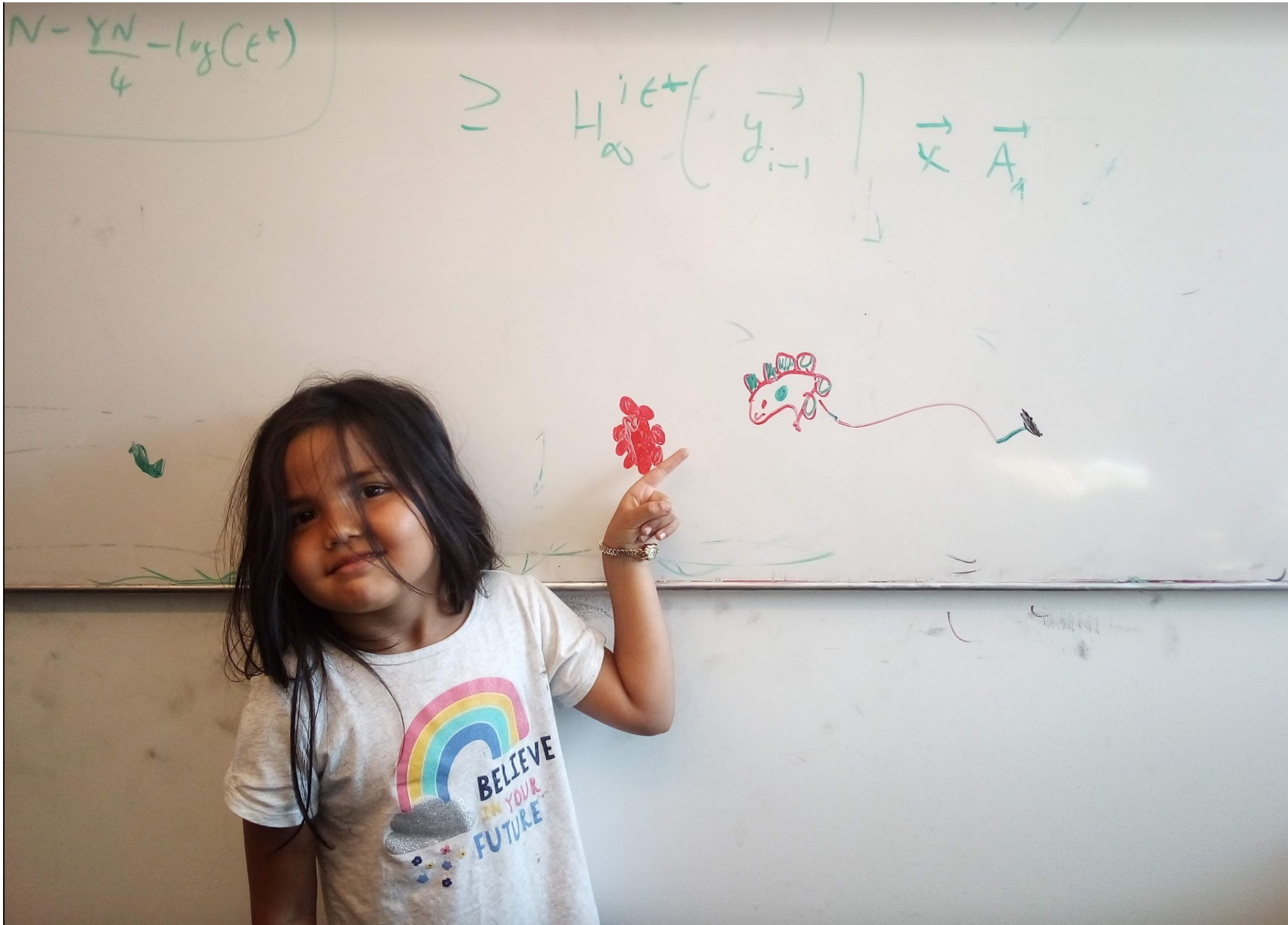
Add i to S — O(1)

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

Return S* = S — O(n)

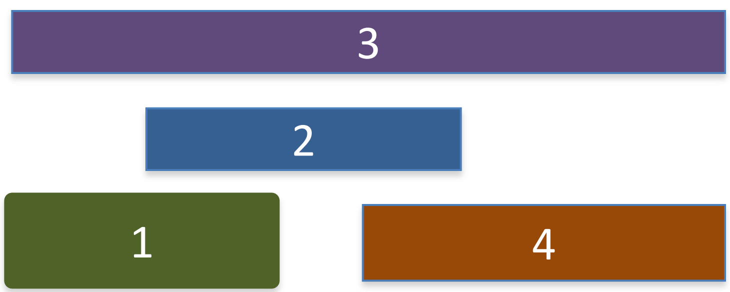
Overall:
O(n) +
n*O(n) =
$O(n^2)$

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 $f(i) \le f(i+1)$

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

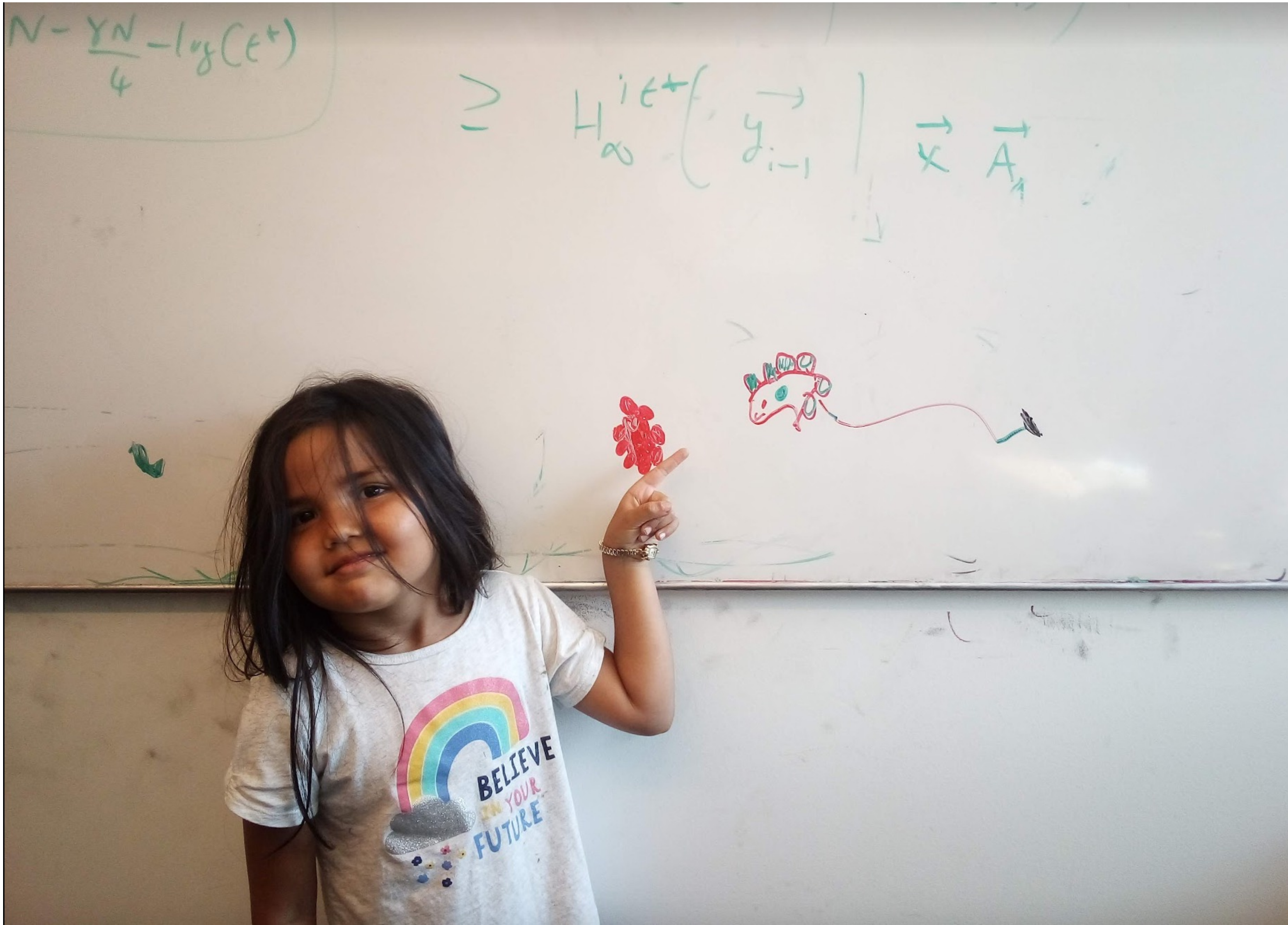
If $s[i] \ge f$

Add i to S

Set f = f(i)

Return S* = S

Questions/Comments?

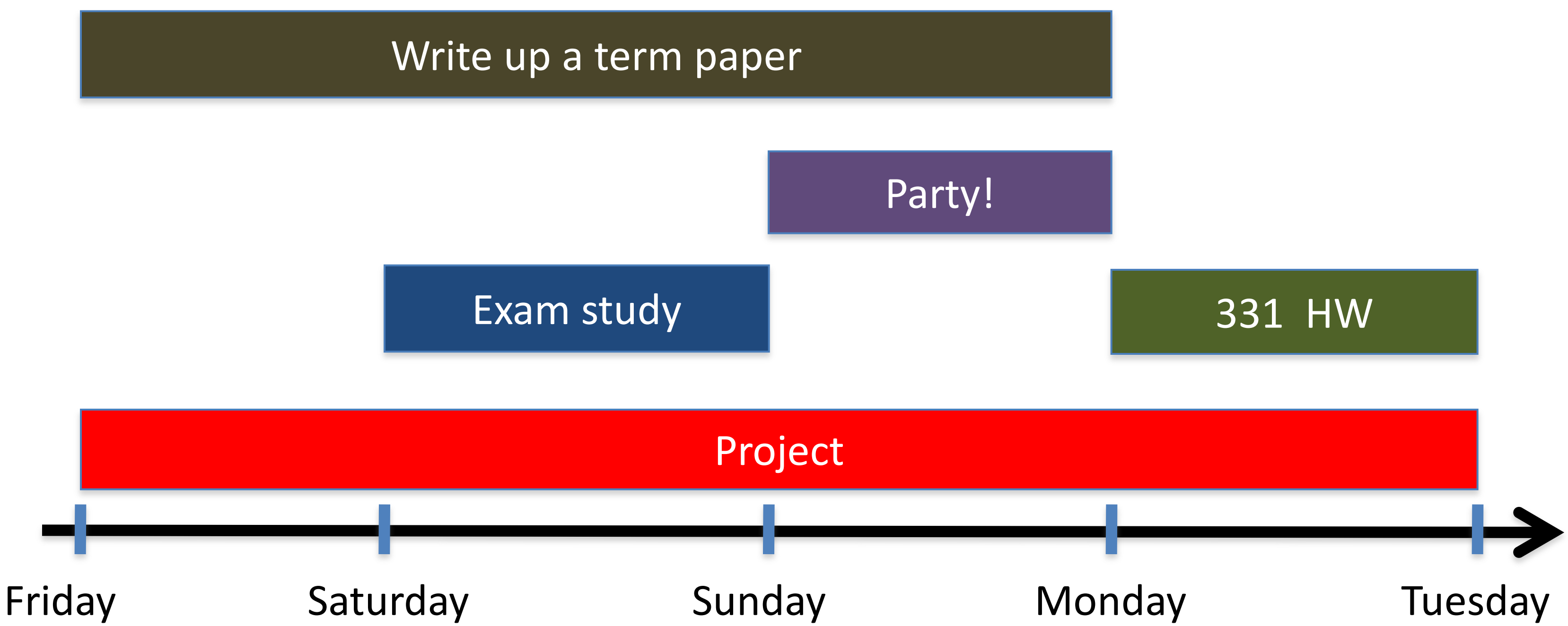


Reading Assignment

Sec 4.1 of [KT]

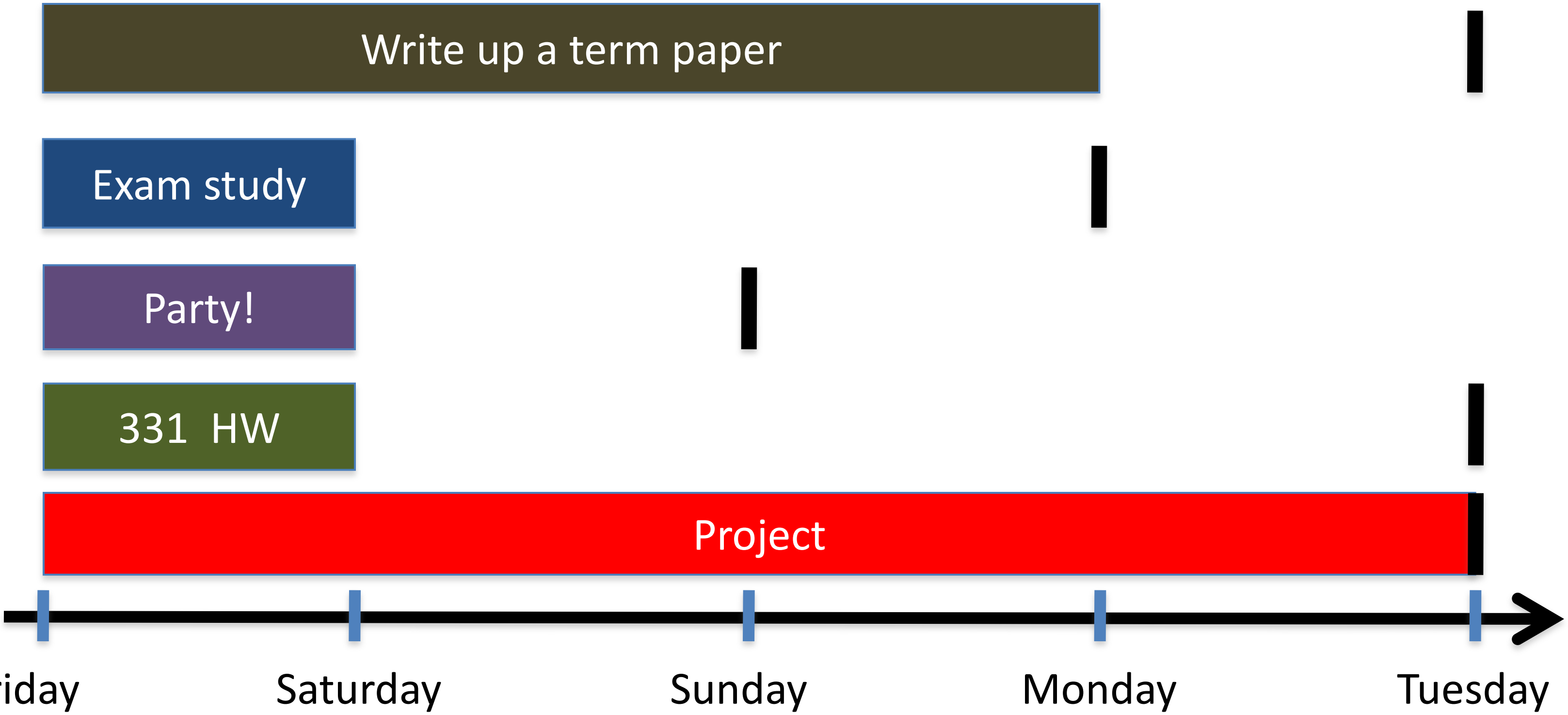
The “real” end of Semester blues

There are deadlines and durations of tasks



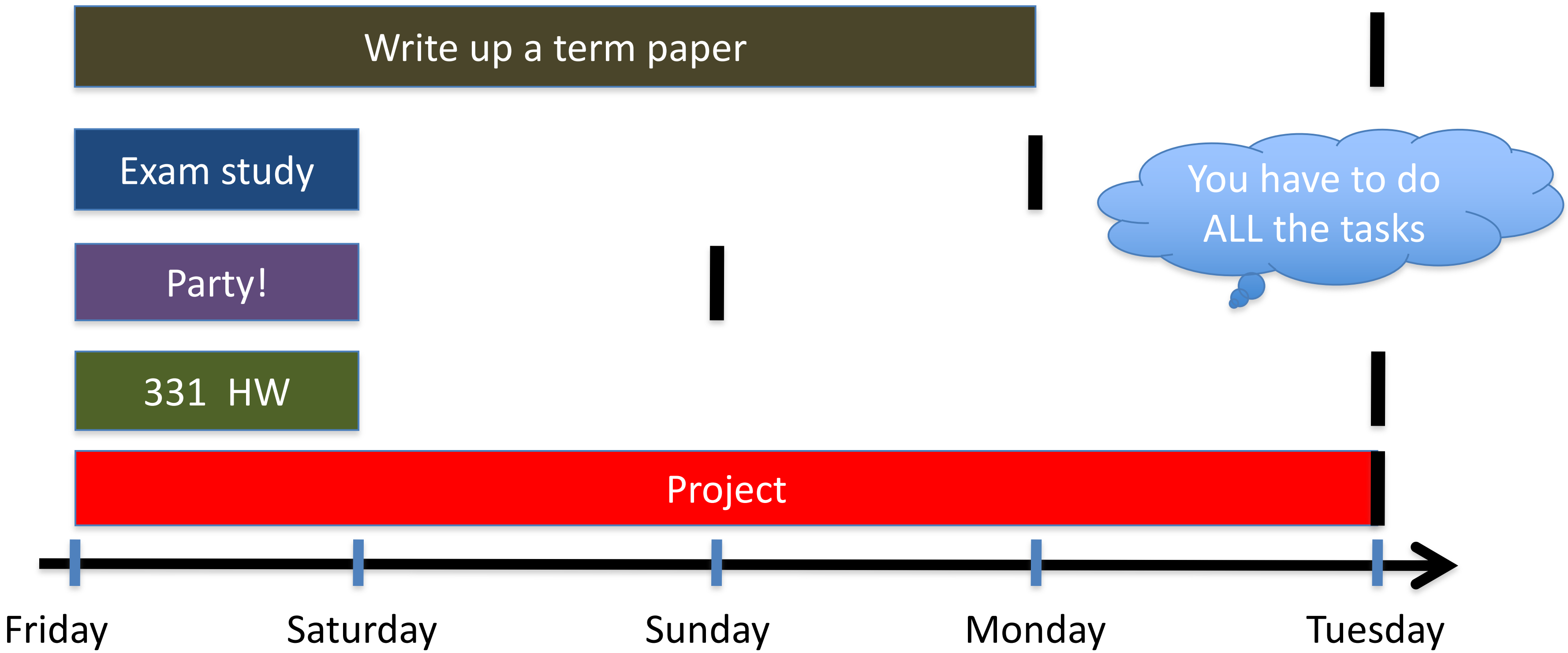
The “real” end of Semester blues

There are deadlines and durations of tasks



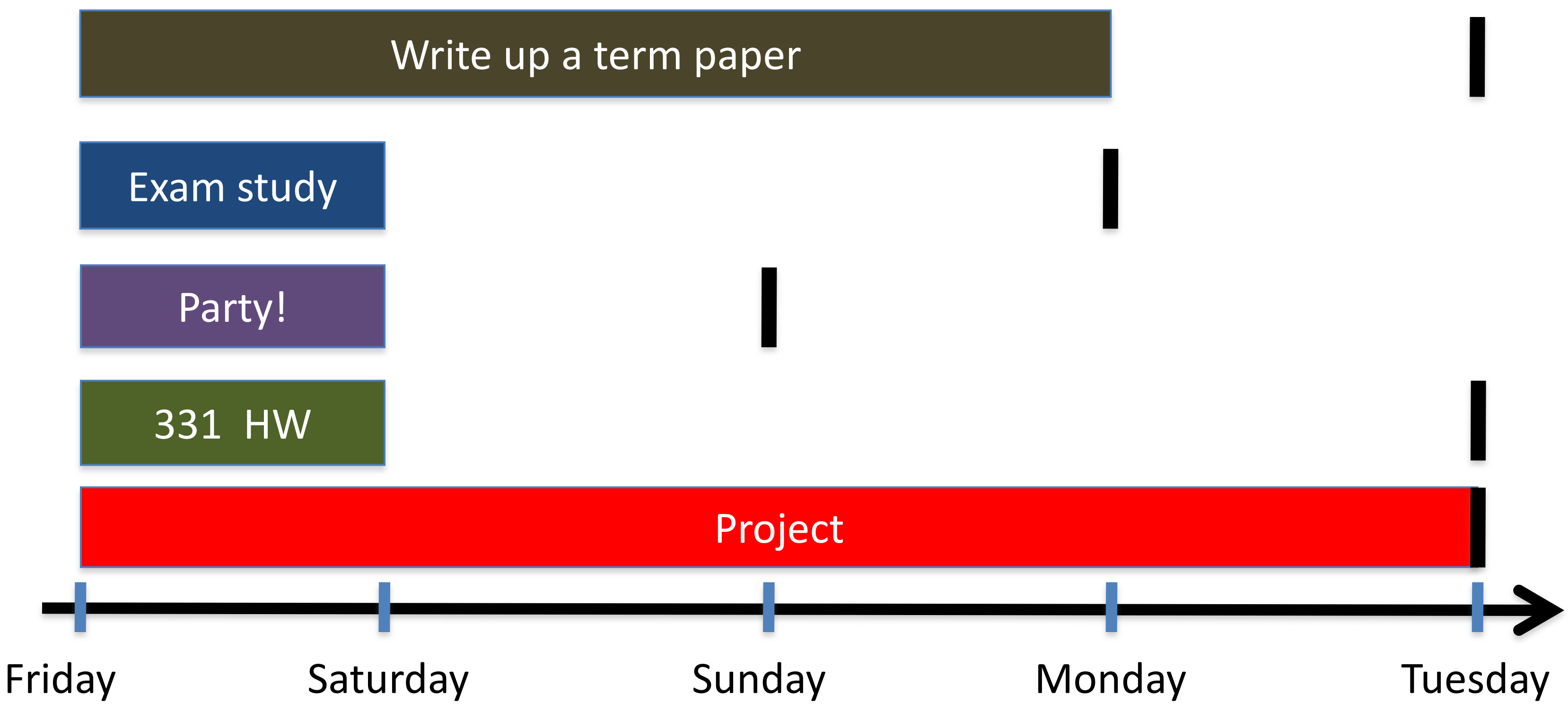
The algorithmic task

YOU decide when to start each task



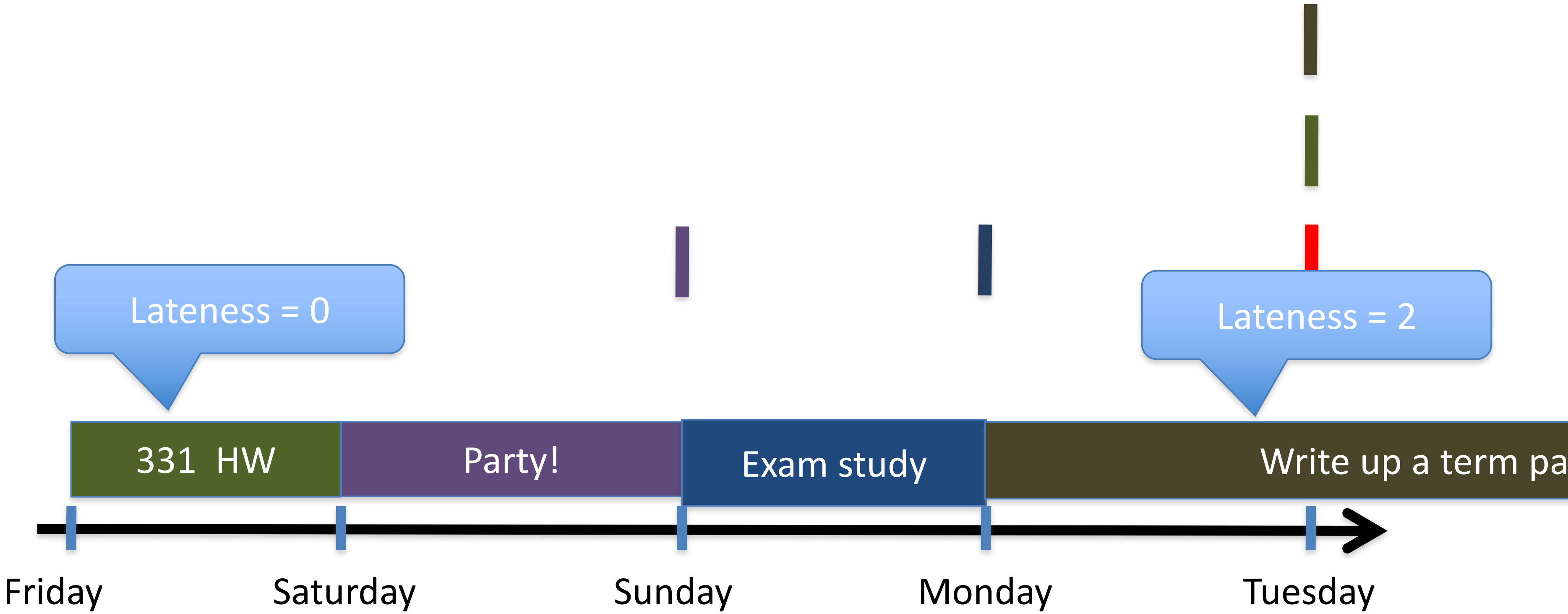
Scheduling to minimize lateness

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.

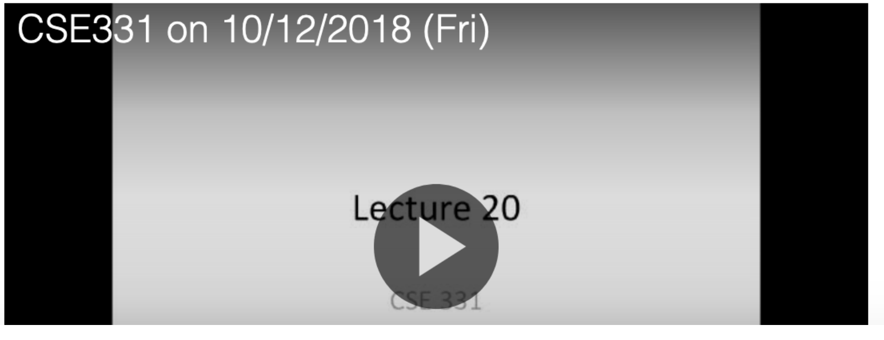
Where does the textbook talk about this?

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

Fall 2018 material

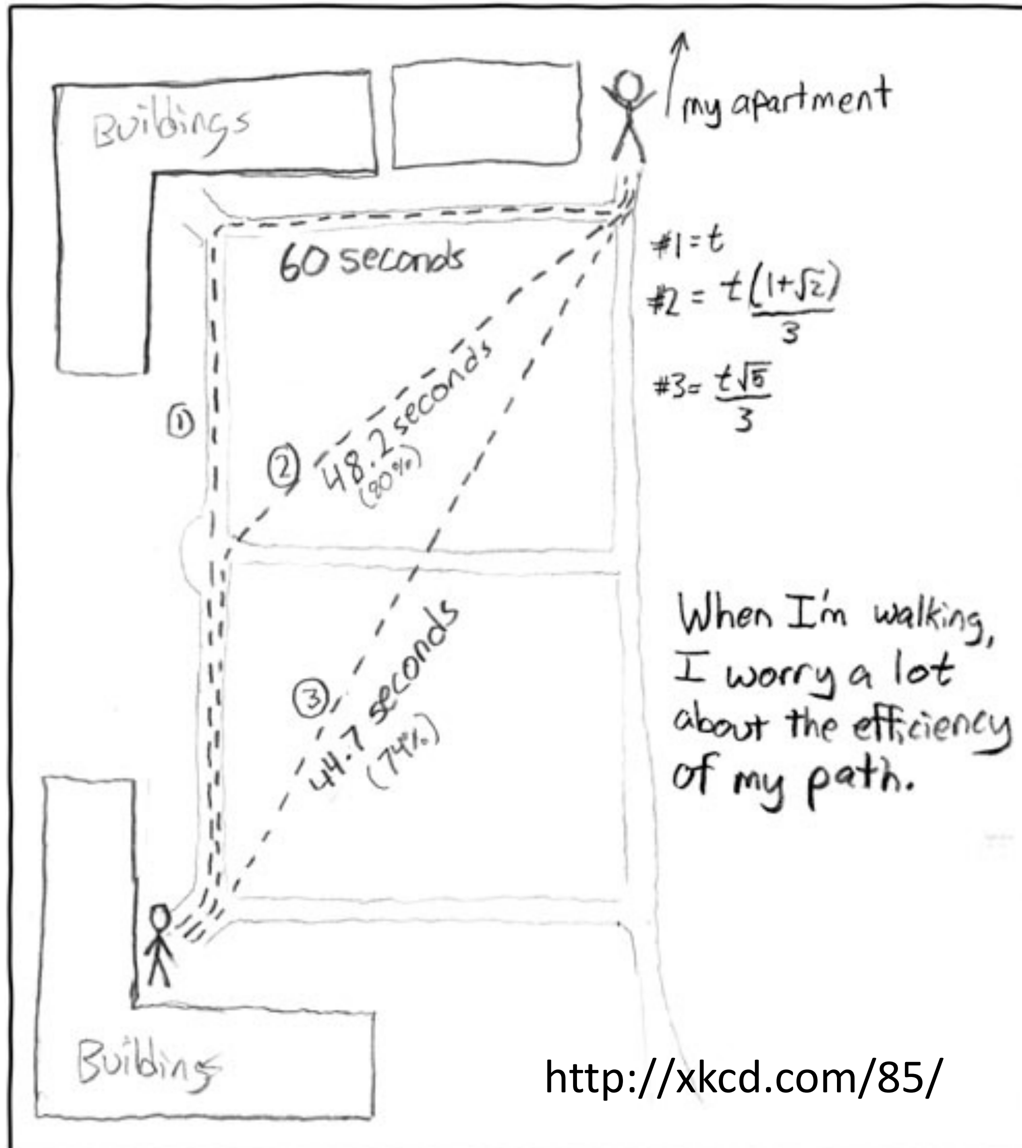
First lecture

Here is the lecture video:



Rest of today

Shortest Path Problem



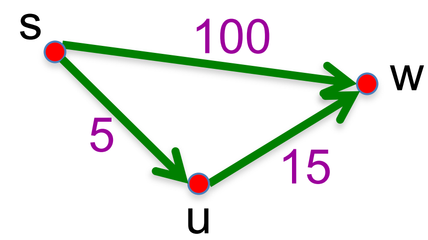
http://xkcd.com/85/

Reading Assignment

Sec 2.5 of [KT]

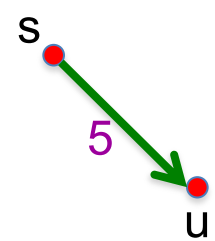
Shortest Path problem

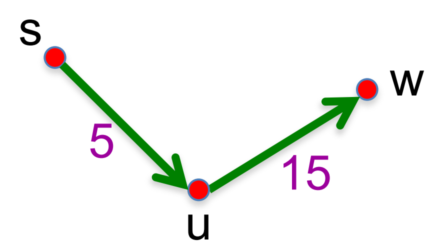
Input: *Directed* graph G=(V,E)



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