Lecture 22

CSE 331

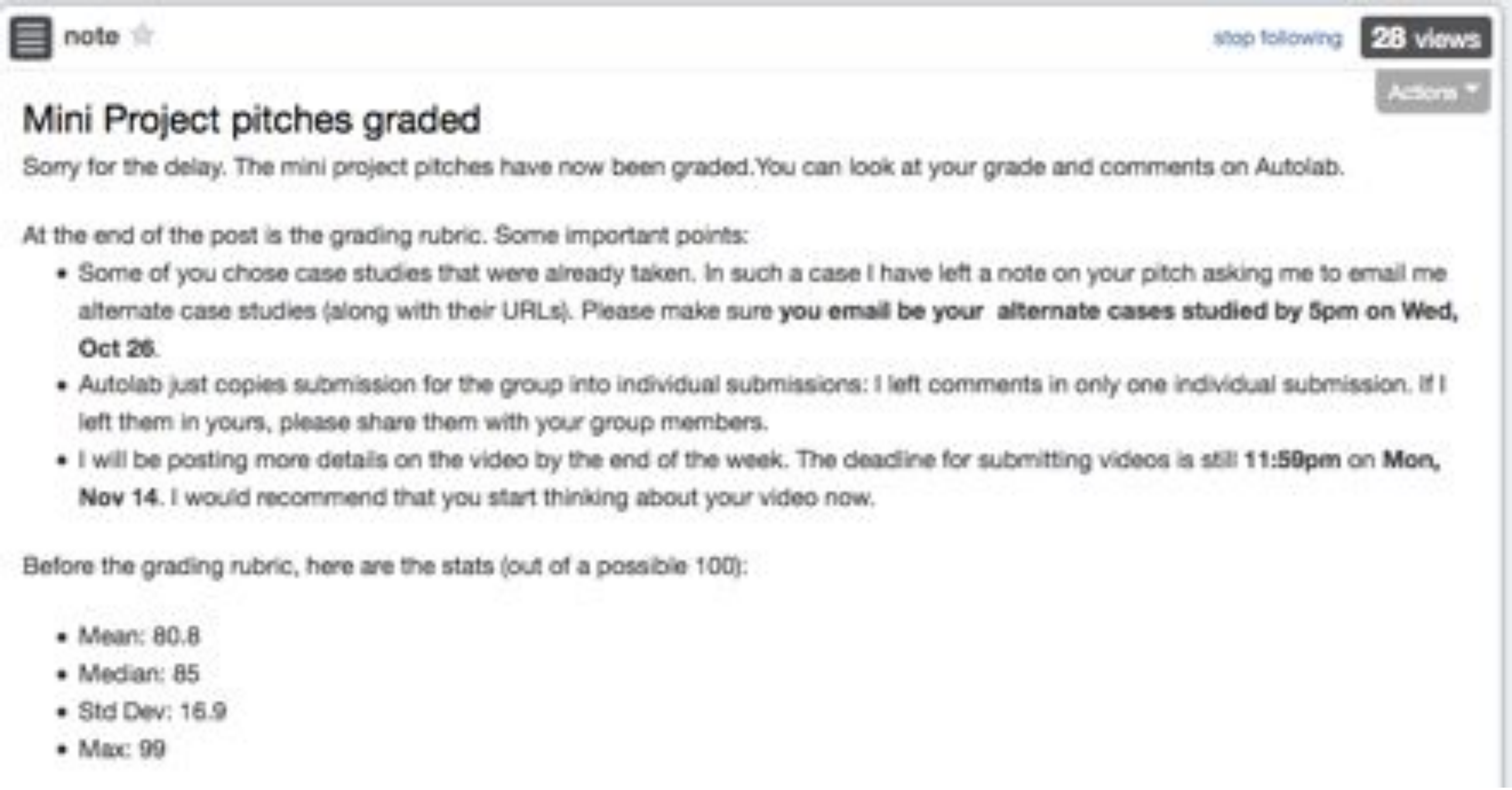
Oct 24, 2016

Graded mid-term-I

Grades for mid-term-I released on Autolab

Should have a post up by tonight on the grading rubric

Graded pitch



Mini Project pitches graded

Sorry for the delay. The mini project pitches have now been graded.You can look at your grade and comments on Autolab.

At the end of the post is the grading rubric. Some important points:

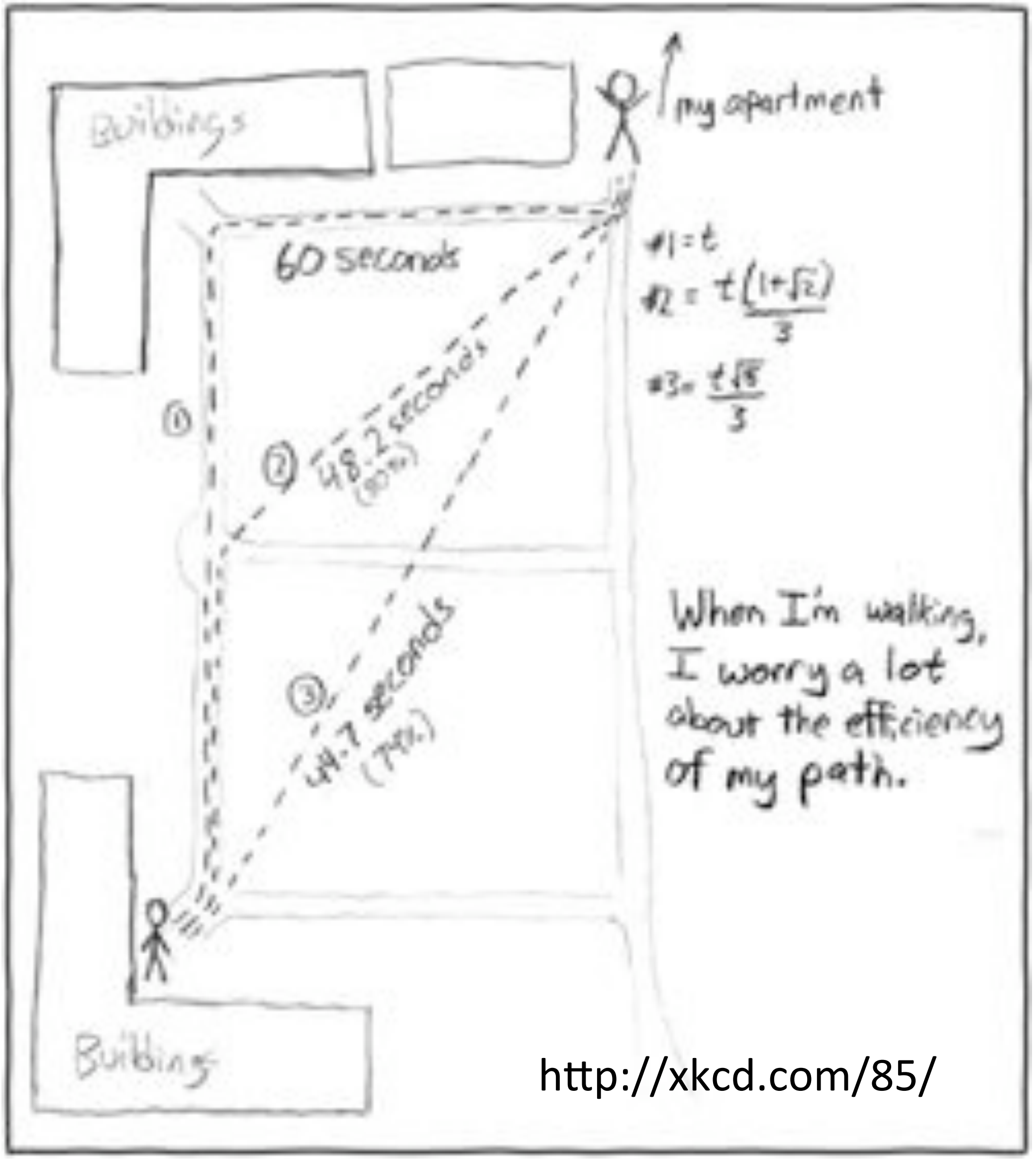
- Some of you chose case studies that were already taken. In such a case I have left a note on your pitch asking me to email me alternate case studies (along with their URLs). Please make sure **you email be your alternate cases studied by 5pm on Wed, Oct 26**.
- Autolab just copies submission for the group into individual submissions: I left comments in only one individual submission. If I left them in yours, please share them with your group members.
- I will be posting more details on the video by the end of the week. The deadline for submitting videos is still **11:59pm** on **Mon, Nov 14**. I would recommend that you start thinking about your video now.

Before the grading rubric, here are the stats (out of a possible 100):

- Mean: 80.8
- Median: 85
- Std Dev: 16.9
- Max: 99

Today

Shortest Path Problem



http://xkcd.com/85/

Reading Assignment

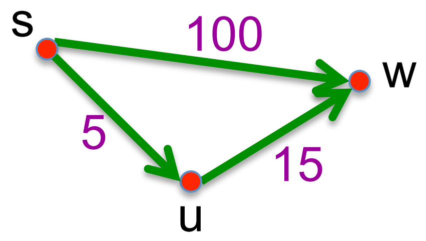
Sec 2.5 of [KT]

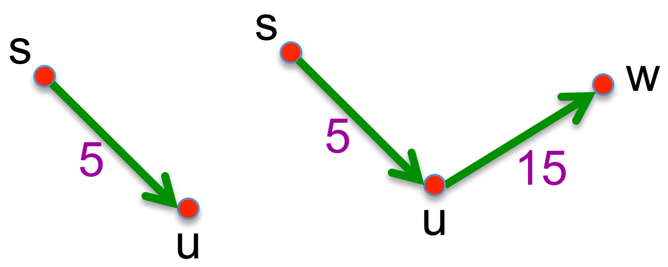
Shortest Path problem

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

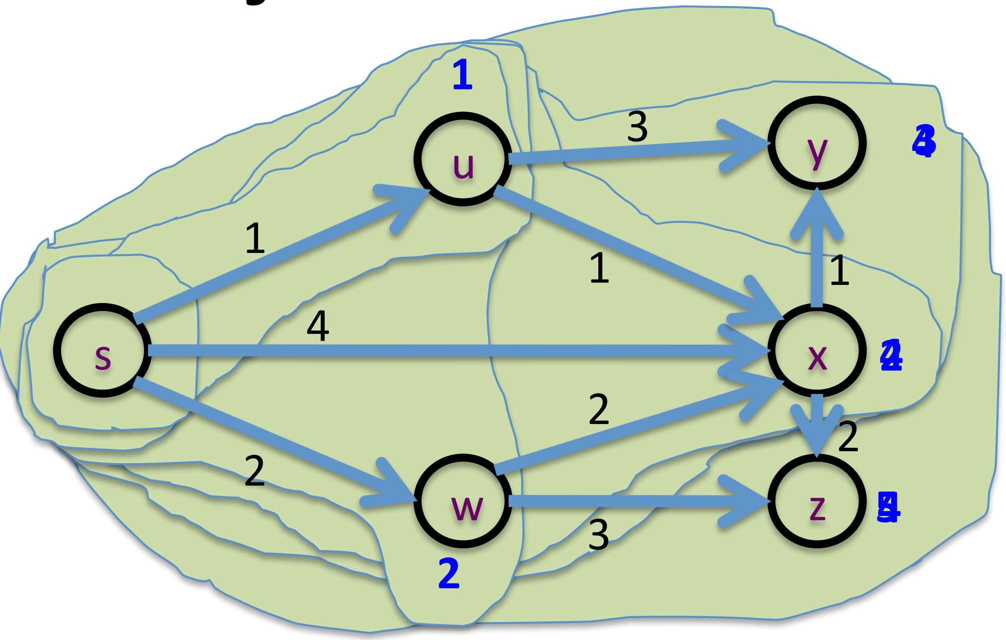
Naïve Algorithm

$\Omega(n!)$ time

Dijkstra’s shortest path algorithm

E. W. Dijkstra (1930-2002)

Dijkstra’s shortest path algorithm



$d'(w) = \min_{e=(u,w) \text{ in } E,\ u \text{ in } R} d(u)+l_e$

d(s) = 0	d(u) = 1
d(w) = 2	d(x) = 2
d(y) = 3	d(z) = 4

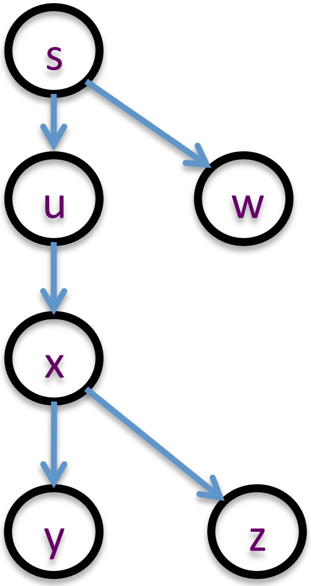
Input: Directed $G=(V,E)$, $l_e \geq 0$, s in V

```
R = {s}, d(s) =0

While there is a x not in R with (u,x) in E, u in R

    Pick w that minimizes d’(w)
    Add w to R
    d(w) = d’(w)
```

Shortest paths



Couple of remarks

The Dijkstra's algo does not explicitly compute the shortest paths

Can maintain "shortest path tree" separately

Dijkstra's algorithm does not work with negative weights

Left as an exercise