Lecture 22

CSE 331

Oct 15, 2014

Graded Quiz 1

Will hand them out at the END of the lecture

note

stop following | 6 views

Actions

Graded Quiz 1

Quiz 1 has been graded and your scores have been uploaded to UBLearns. I'm hoping to hand out the quizzes in class tomorrow (if not in the worst-case they'll be available for pickup at the extra office hours tomorrow @232).

First the stats:

- Mean: 3.83
- Median: 4.5
- Std Dev: 2.28

Now to the grading rubric. First the rubric that is applicable to both Q1a and Q1b on the quiz:

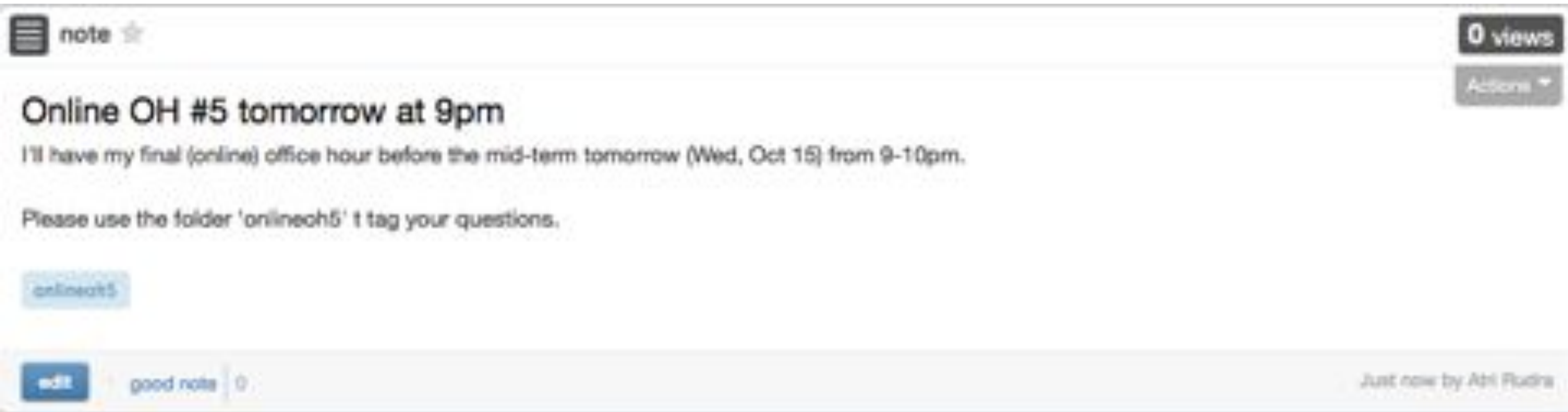
1. Two points for getting the T or F correct (which in this quiz was always true).
2. Three points for a correct justification.
3. You get zero points if you state the T or F incorrectly (irrespective of what you stated in the justification part).

Note that we will also be using the same general rubric as above for the questions on the first mid-term.

Now for some more specific guidelines/mistakes for the justification parts:

- For Q 1a:
    1. For a technically correct solution your algorithm should work for all graphs even if they are disconnected. However, you got full points for the justification part if your justification only worked for the case when G is connected.
    2. If you mentioned DFS or BFS but rest of the stuff was wrong you got 1 out of 3

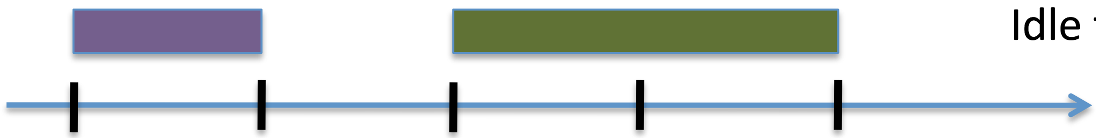
Online OH tonight @9pm



Two definitions for schedules

Idle time

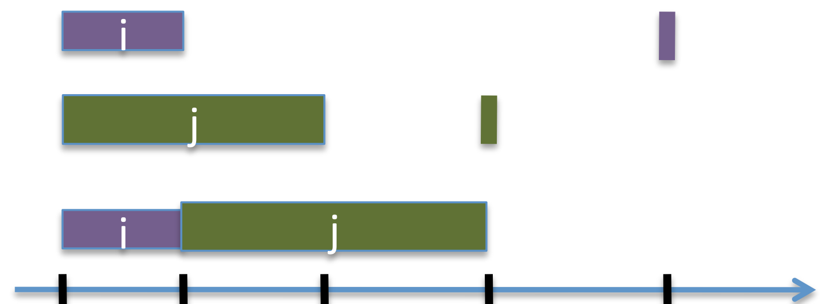
Max "gap" between two consecutively scheduled tasks



Idle time =0

Inversion

(i,j) is an inversion if i is scheduled before j but $d_i > d_j$



f=1

For every i in 1..n do

Schedule job i from s_i=f to f_i=f+t_i

f=f+t_i

0 idle time and 0 inversions for greedy schedule

Proof structure

Any two schedules with 0 idle time and 0 inversions have the same max lateness

Greedy schedule has 0 idle time and 0 inversions

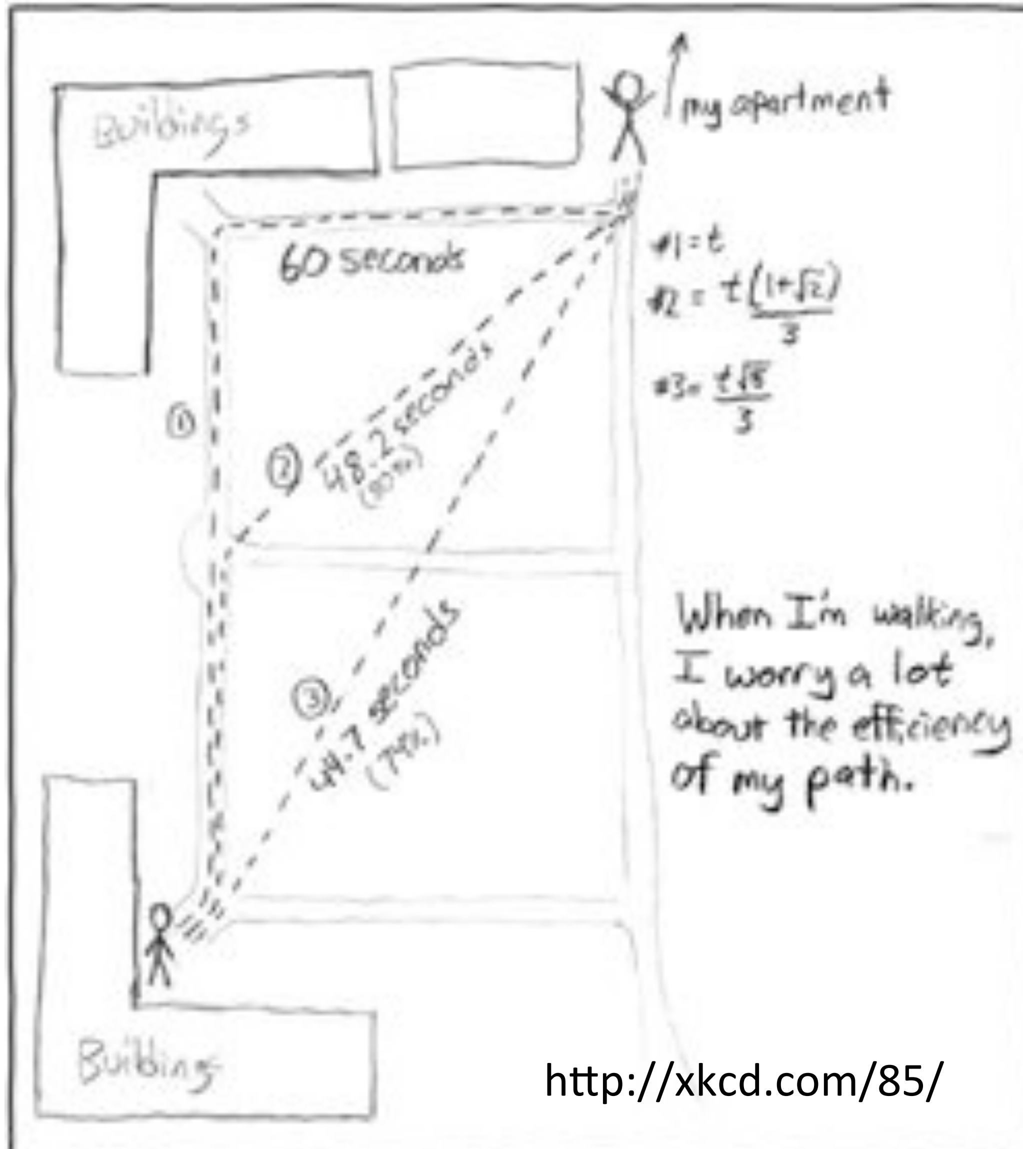
There is an optimal schedule with 0 idle time and 0 inversions

Today's agenda

"Exchange" argument to convert an optimal solution into a 0 inversion one

Rest of Today

Shortest Path Problem



http://xkcd.com/85/

Reading Assignment

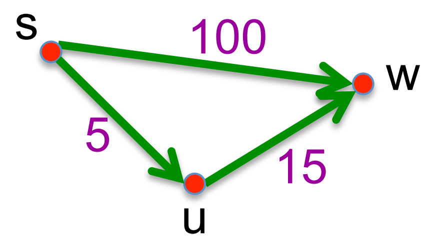
Sec 2.5 of [KT]

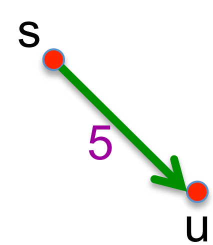
Shortest Path problem

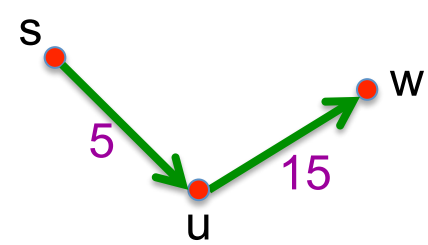
Input: *Directed* graph G=(V,E)

Edge lengths, l_e for e in E

“start” vertex s in V







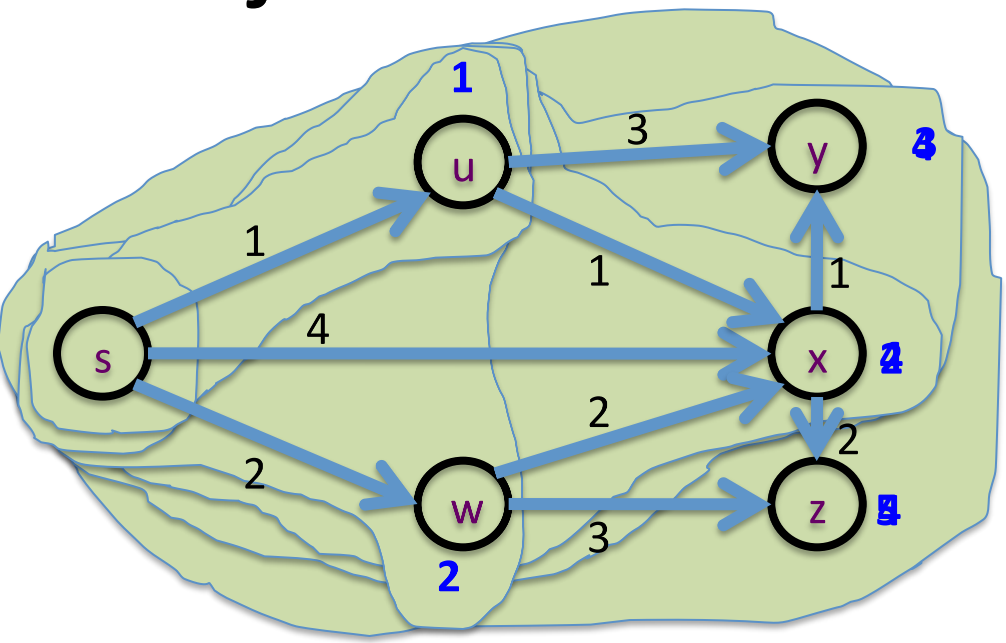
Output: All shortest paths from s to all nodes in V

Naïve Algorithm

$\Omega(n!)$ time

Dijkstra’s shortest path algorithm

Dijkstra's shortest path algorithm

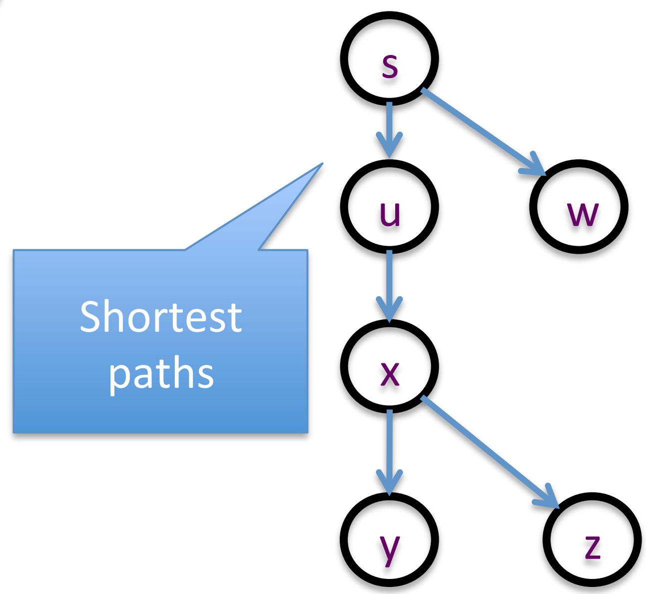


$d'(w) = \min_{e=(u,w) \text{ in } E,\, u \text{ in } R} d(u)+l_e$

$d(s) = 0$ $d(u) = 1$

$d(w) = 2$ $d(x) = 2$

$d(y) = 3$ $d(z) = 4$



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

$d(w) = d'(w)$

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