Lecture 21

CSE 331

Oct 21, 2016

Grading

Mid-term-1 hopefully by today

Mini project pitch by the weekend

Scheduling to minimize lateness

n jobs: ith job (t_i, d_i)

start time: s

Schedule the n jobs: ith job gets interval $[s(i), f(i)=s(i)+t_i)$

Algo picks s(i)

GOAL: Minimize MAXIMUM lateness

Lateness of job i, $l_i = \max(0, f(i)-d_i)$

The Greedy Algorithm

(Assume jobs sorted by deadline: $d_1 \leq d_2 \leq \ldots.. \leq d_n$)

f=s

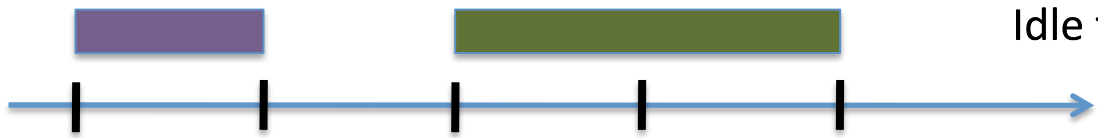
For every i in 1..n do

Schedule job i from $s(i)=f$ to $f(i)=f+t_i$

$f=f+t_i$

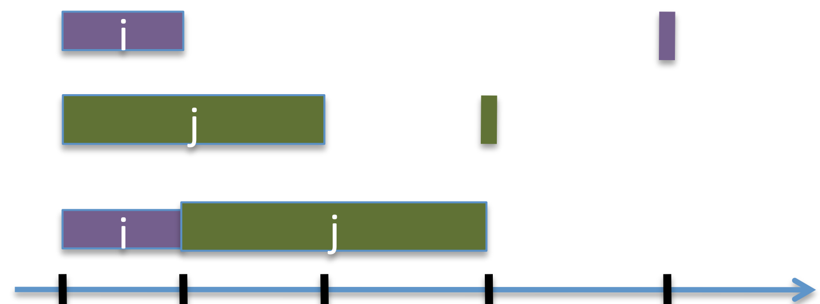
Two definitions for schedules

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

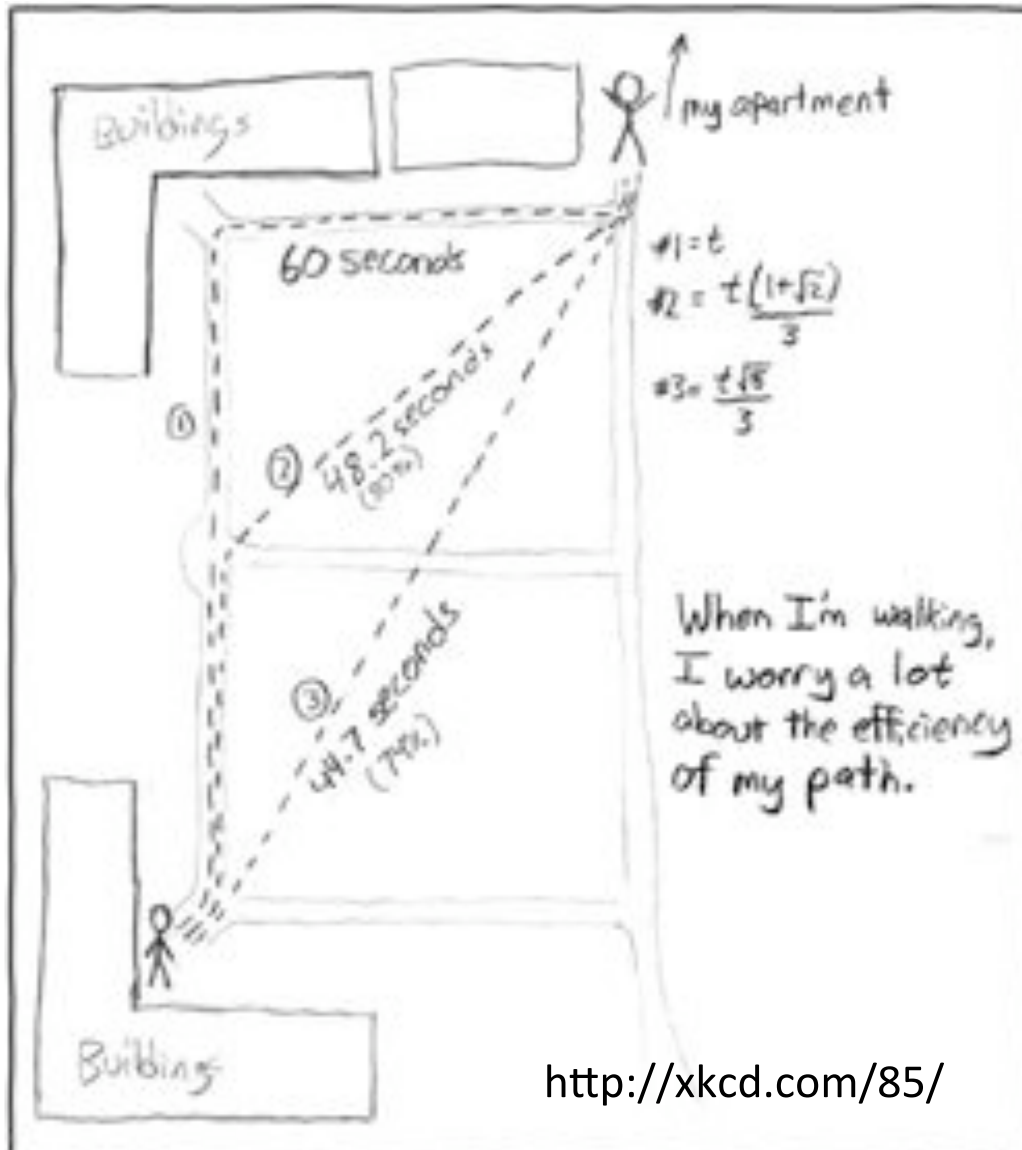
There is an optimal schedule with 0 idle time and 0 inversions

Today's agenda

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

Rest of Today

Shortest Path Problem

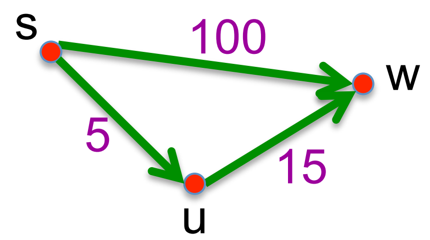


http://xkcd.com/85/

Reading Assignment

Sec 2.5 of [KT]

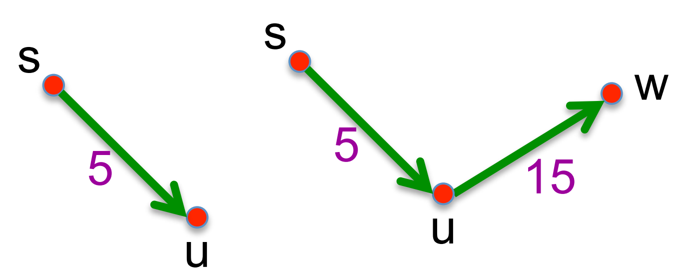
Shortest Path problem



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