CS314 - Greedy Algorithms

Note Title 2/10/2010

Announcements

- HW due Friday
- Office hours tomorrow 9-10am

Scheduling to minimize lateness

Single resource + set of n requests for resource
(like last time)

But here, request i has: $\{1, .., n\}$
deadline d_i
time t_i to run

(Think jobs on a computer requesting processor time)

Many different ways to optimize

Here: Allowed to run past deadline, but want to minimize the maximum lateness

Formally: Assign s_i to each i

Let $f_i = s_i + t_i$

Then lateness $l_i = f_i - d_i$

Want minimize $\max_i l_i$

Example:

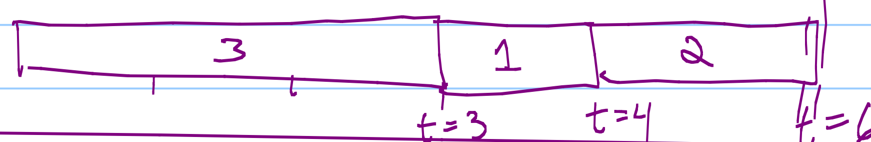
Job 1 — length 1 — $d_1 = 2$

Job 2 — length 2 — $d_2 = 4$

Job 3 — length 3 — $d_3 = 6$

How can we schedule?

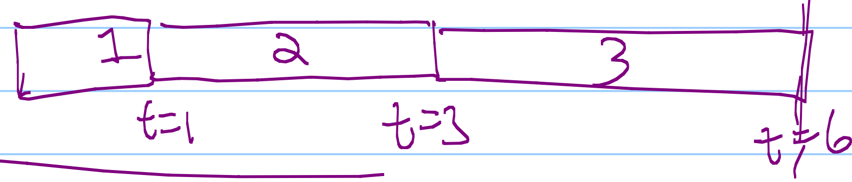
$\boxed{2}$



$l_1 = 2$

$l_2 = 2$

$\boxed{0}$



$l_1 = 0$

$l_2 = 0$

$l_3 = 0$

Greedy Strategies?

- Shortest to longest t_i

  1: $t_1 = 6$ $d_1 = 6$

  2: $t_2 = 1$ $d_2 = 9$

- Shortest slack time $d_i - t_i$

  1: $t_1 = 10$ $d_1 = 10$ (slack $= 0$)

  2: $t_2 = 1$ $d_2 = 2$

Instead - earliest deadline first (EDF)

Sort by d_i (& reorder t_i accordingly)
$f \leftarrow 0$
for $i \leftarrow 1$ to n
    $s[i] \leftarrow f$
    $f \leftarrow s[i] + f[i]$
return $S[1..n]$

$O(n)$

Runtime? $O(n \log n)$

Note - no inverted pairs in our algorithm.

job i runs before job j but $d_i > d_j$

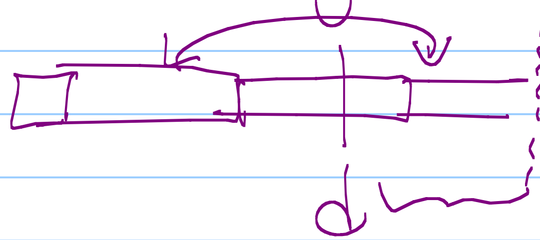
Proof of correctness:

⇒ **Lemma:** All schedules with no inversions and no idle time have same maximum lateness.

pf: Two schedules, both have no inversions & no idle time.

Only possible difference is jobs with same deadline in different order.

So consider all jobs w/ a deadline d.



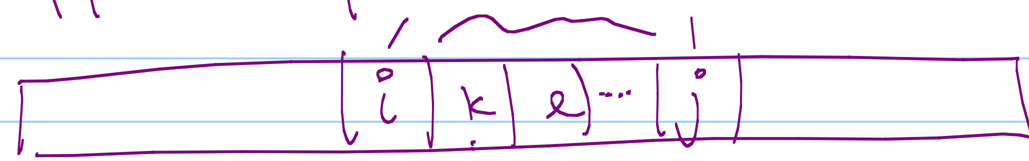
no matter of order, last one finishes at same time so lateness is same. ∎

Lemma 2: There is an optimal schedule with no idle time.

(obvious)

→ **Lemma:** There is an optimal schedule with no inversions (& no idle time).

pf: Suppose optimal schedule $\mathcal{O}$ has inversions.



$d_i > d_j$

Want to find an adjacent inversion →

If there is inverted pair i, j
can step up job by job from i.
If find adjacent pair along way, done.
If not, eventually get to j → must have reached adjacent inversion.

a | b   nothing changes

$d_a > d_b$

Swap a + b: fewer inversions
haven't changed any lateness
besides a + b.   finish − deadline

Change only f_a + f_b.
a finishes at "old" f_b
b finishes at time f_a

$f_b > f_a$

Had: $f_a - d_a$ $\quad$ $d_a > d_b$

$f_b - d_b$

Now: $f_b - d_a$ $< f_a - d_a$

$f_a - d_b$

~~Reword: a now finishes at f_b~~

so its lateness is $f_b - d_a$

We had $f_b - d_b$ and $d_a > d_b$

So a can't be more late in new schedule than b was in old.

since $f_b - d_a < f_b - d_b$ ←

End of argument:

Swapping inverted pairs can not hurt.

Take $\mathcal{O}$ & start swapping.

at most $O(n^2)$ inverted pairs.

$\Rightarrow$ end with optimal schedule w/ no inverted pairs (& no idle time)

apply lemma 1.

$\blacksquare$