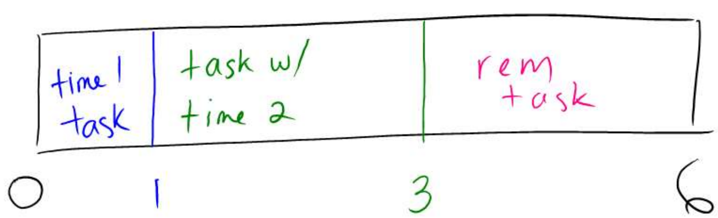
CompSci 161
Spring 2021 Lecture 21:
Greedy Algorithms:
Scheduling with Deadlines

2

Scheduling with Deadlines

Example 1: What is the optimal schedule for the following input?

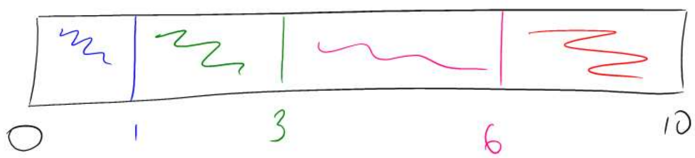
Time	1	2	3
Deadline	2	4	6



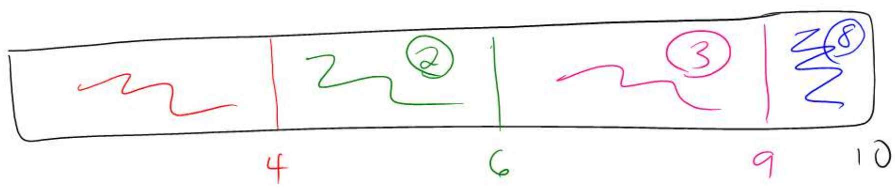
3 Scheduling with Deadlines

Example 2: What is the optimal schedule for the following input?

Time	1	2	3	4
Deadline	2	4	6	6



penalty: 4



4 Possible Scheduling Algorithms

- Sort the jobs by increasing time t_i; schedule them in that order.

Time	1	2
Deadline	100	2

OPT: 0
(better)

ALG: 1 penalty
1 then 2

- Sort the jobs by $d_i - t_i$; schedule them in that order.

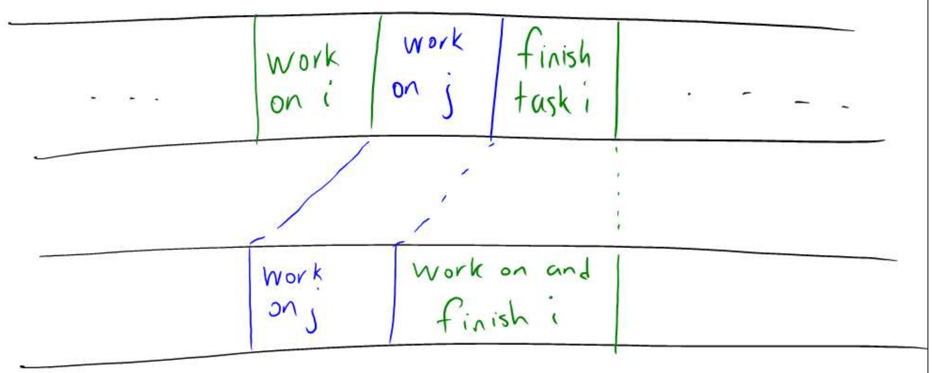
Time	6	1
Deadline	7	3

ALG:
6 then 1
Penalty: 4

better: 1 then 6
Penalty: 0

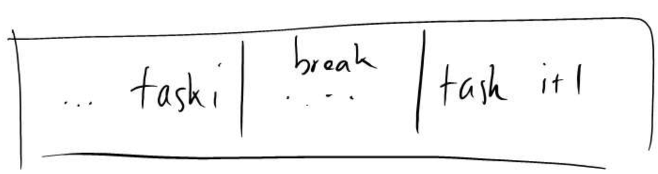
Can we break up tasks?

Is it beneficial to break up tasks? Why or why not?



Proof: Lemma 1

When deciding start times, don't leave any gaps; $s_{i+1} = s_i + t_i$.



7

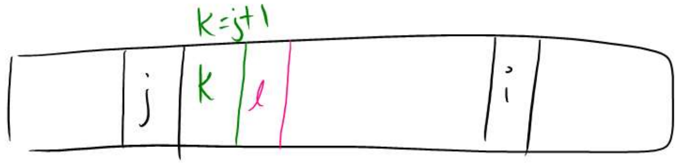
Proof: Lemma 2

A: my order w/ i<j

Yours: | | j | i |

Any schedule that doesn't agree with our algorithm has at least one pair of *consecutive* intervals $i, i+1$ that are *inverted* relative to our order.

k=j+1



Are j,k inverted?

→ IF SO, yes. I found consec inverted pair :)

ELSE? $\underbrace{A_j < A_k}$, $\underbrace{A_i < A_j}$

So I have $A_i < A_k$

(i,k) are inverted

8

We can now finish the proof

Algorithm: schedule by increasing d_i

Claim: Any schedule with an inversion can be modified to be more like our algorithm's output without making it worse. Suppose X: any other perm

f_i': finish of i after the swap (i.e., in ALT)

By lemma 2, $\exists i,j$ $d_i > d_j$ and $j = i+1$

Let S_i = start of i in X

f_i = finish i = $S_i + t_i$ $\quad f_j = \overbrace{S_i + t_i}^{S_j} + t_j$

f_j' similar

ALT = X w/ i,j swapped

$f_i' = S_i + t_j + t_i = f_j$ $\qquad f_j' = S_i + t_j$

Penalty worse?

$d_i > d_j$ lateness i' in ALT $\leq$ lateness j in X

penalty $j' \leq$ penalty j in X

$\therefore$ ALT is no worse than X

9

Proof of Correctness

- We proved this:
  Claim: Any schedule with an inversion can be modified (by removing an adjacent inversion) to be more like our algorithm's output without making it worse.

- What does the full proof look like?

  "BubbleSort" Comparison
  (see end of lec vid)