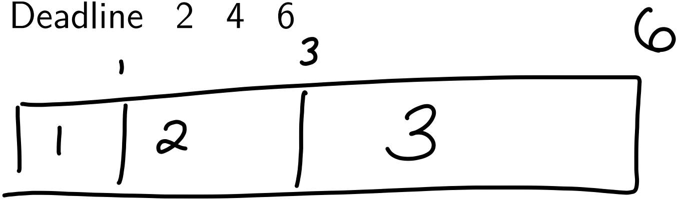
CompSci 161
Winter 2023 Lecture 18:
Greedy Algorithms:
Scheduling with Deadlines

Scheduling with Deadlines

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

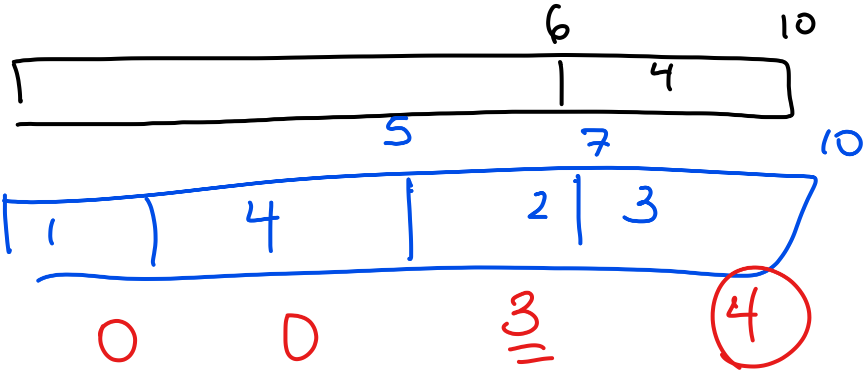
Time	1	2	3
Deadline	2	4	6



Scheduling with Deadlines

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

Time	1	2	3	4
Deadline	2	4	6	6



Possible Scheduling Algorithms

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

t :	1	2	3	1000
d :	1050	1050	1050	1000

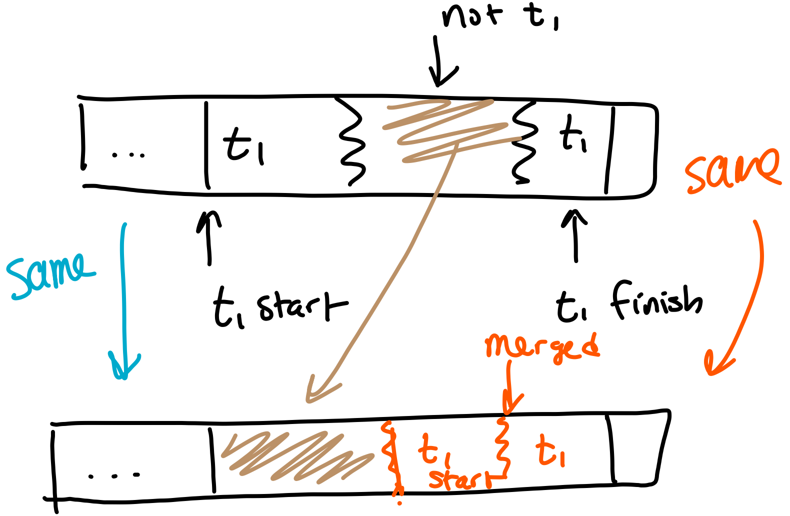
- Sort the jobs by $d_i - t_i$; schedule them in that order.

"slack"

2	1000
1	1000

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; $s_{i+1} = s_i + t_i$.

Proof: Lemma 2

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.

$\exists\, i,j$ s.t. $i<j$ but $A[i]>A[j]$

if $j=i+1$ done

else

let $k=i+1$.

$A[i], A[k]$ inverted?

Yes? done

No? $A[i]<A[k]$ and $A[i]>A[j]$

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.

i,j: adj inverted: $i<j$ but ?? $d_i \geq d_j$

swap tasks i,j

$f_i = s_i + t_i$

$f_j = s_i + t_i + t_j$

$f_i' = s_i + t_j + t_i$

no worse iff $d_i \geq d_j$

$f_j' = s_i + t_j < f_j$

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?