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

<table>
<thead>
<tr>
<th>Time</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deadline</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

[Diagram of schedule]
Example 2: What is the optimal schedule for the following input?

<table>
<thead>
<tr>
<th>Time</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deadline</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>
Possible Scheduling Algorithms

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

\[ t: \quad 1 \quad 2 \quad 3 \quad 1000 \]
\[ d: \quad 1050 \quad 1050 \quad 1050 \quad 1000 \]

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

2

\[ d: \quad 1050 \quad 1050 \quad 1050 \quad 1000 \]
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?

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.

\[
\begin{align*}
i,j & : \text{adj inverted: } i < j \\
f_i = S_i + t_i \\
f_j = S_i + t_i + t_j
\end{align*}
\]

\[
\begin{align*}
\text{swap tasks } i,j \\
\text{no worse iff } d_i \geq d_j
\end{align*}
\]

\[
\begin{align*}
f_i' = S_i + t_j + t_i \\
f'_j = S_i + t_j < f_j
\end{align*}
\]
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?