Greedy Algo

0. (Sort so that $d_1 \le d_2 \le \dots \le d_n$) $\leftarrow O(n \log n)$
1. $f \leftarrow s$ $O(1)$
2. for $i = 1 \dots n$ $O(n)$

overall: $O(n \log n)$

$O(1)$ { Schedule i in the interval $[s(i) = f, f(i) = f + t_i)$
$O(1)$ { $f \leftarrow f + t_i$

Prop: Output of the greedy algo has 0 idle time & 0 #inversions

Def: Idle time: $\max_{i=1 \dots n-1} s(i+1) - f(i)$

Def: Inversion: Pair (i, j) is an inversion (w.r.t. a given schedule) if with respect to i is scheduled before j but $d_i > d_j$

Thm: The output of the greedy algorithm is optimal.

Pf. idea:

Lemma 1: If two (valid) schedules both have 0 idle time & 0 # inversion, then both have the same max. lateness.

Lemma 2: There exists an optimal schedule with 0 idle time & 0 #inversions

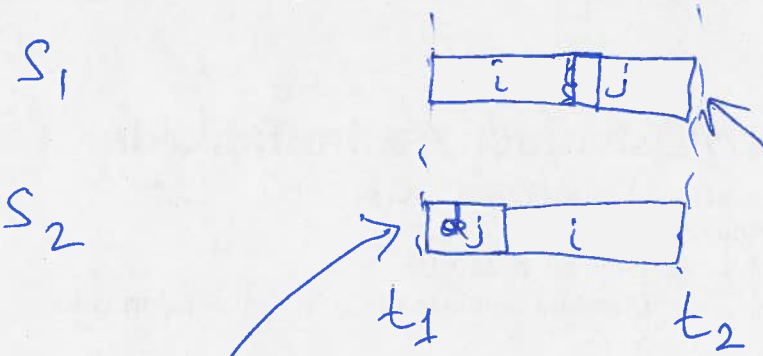
Pf (sketch) of Lemma 1:

Let S_1 & S_2 be schedules w/ 0 idle time & 0 # inversions.

Obs: All jobs j w/ $d_j < d_i$ get scheduled before d_i

(As both have 0 # inversions)

$d < d' < d''$



as S_1 & S_2 have 0 idle time (+ 0 # inv.)

max lateness $= \max(0, t_2 - d')$ $=$ max lateness fd d' guys $\max(0, t_2 - d')$

$\Rightarrow$ max lateness of S_1 = max lateness of S_2.