Thm

Greedy (earliest finish time) is optimal for interval scheduling.

Proof

Let \( i_1, i_2, \ldots, i_k \) be jobs greedily picked.
Let \( j_1, j_2, \ldots, j_m \) be any valid solution. (think \( j_1 \) as OPT)

Claim: For all \( i, j \), \( f(i) \leq f(j) \)

Proof by induction

Base case: \( f(i_1) \leq f(j_1) \) (Greedy picks the job finish first)

IH: \( f(i_k) \leq f(j_k) \)

IS: \( f(i_{k+1}) \leq f(j_{k+1}) \)

\( f(i_{k+1}) \leq f(j_{k+1}) \leq f(j_{k+1}) \)

So, \( j_{k+1} \) is one of candidate greedy consider.

Since greedy picks with finish time.

\( f(i_{k+1}) \leq f(j_{k+1}) \)

why \( k \geq m \)?
why \( K \geq m \)?

Consider step \( m-1 \).

\[ J_i < J_{m-1} \]

so, job \( j_{m-1} \) is again candidate.

so, greedy must pick some job.

---

**Lemma:** Swapping 2 adjacent inverted jobs does not \( \geq \) max \( L \).

**Proof**

before:

\[ \begin{array}{cccc}
| & J & I & \uparrow f_i & \end{array} \]

after swapping:

\[ \begin{array}{cccc}
& I & J & \uparrow f_i & \end{array} \]

Note \( L_i = \begin{cases} 0 & \text{if } f_i \leq d_i \\ f_i - d_i & \text{otherwise} \end{cases} \)

Let \( L'_i \) is the lateness after swapping

\[ L'_i = \begin{cases} L_k & \text{if } k \leq \text{max } L \text{ and } (d_i < d_j) \\ L_i & \text{otherwise} \end{cases} \]

\[ L'_i \leq L_i \leq \text{max } L \]

\[ L'_i = f_i - d_j = f_i - d_j < f_i - d_i = L_i \leq \text{max } L \]

\( \Rightarrow \) \( \text{max } L' \leq \text{max } L \)
Lemma 2: There is an optimal schedule if jobs are sorted according to deadline in descending order.
Proof: bubble sort + item 1.

Thus, the greedy is optimal.
Proof: The schedule found in Lemma 2 is greedy. They only differ for jobs with same deadline, but they don’t affect max L. So greedy is optimal.