**Thm.** Greedy is optimal for job scheduling. [Schedule max \( n \) of compatible jobs.]

\[ f(i_1) \leq f(i_2) \leq \cdots \leq f(i_m). \]

Greedy

Goal: \( k \geq m \). [Careful, \( \text{OPT} \) is not necessarily unique.]

**Lem:** For all \( r \geq 1 \), \( f(C_r) \leq f(C_r') \).

**Pf of Lem:** Use induction.

**Base Case** \( r=1 \). First job in greedy has smallest finishing time.

**I.** Supp for some \( r \geq 1 \), \( f(C_r) \leq f(C_r') \).

**II.** To show \( f(C_{r+1}) \leq f(C_{r+1'}) \).

We know \( f(C_r) \leq f(C_r') \leq s(i_r) \). \( i_r \) is compatible with \( C_r' \).

So \( i_r \) is a feasible option for Greedy when scheduling \( i_r \) :

\[ f(C_{r+1}) \leq f(C_{r+1'}). \]

**III.** If \( k < m \), \( J_{k+1} \) is compatible with \( i_k \). BC \( f(i_k) \leq f(i_k') \leq s(J_{k+1}) \).

So Greedy should schedule \( J_{k+1} \). Contradiction!

**Thm.** Greedy uses minimum \( \# \) of classes.

**Pf.** Supp Greedy uses \( d \) classes.

Goal: \( d \leq \text{depth} \leq \text{OPT} \)

To show, enough to exhibit a vertical line which crosses \( d \) classes.

Look at time \( t \) that greedy allocates \( d \)-th classroom.

\[
\text{Direction: } \text{Every ALG} \geq \text{depth uses classrooms}
\]
At this time we had allocated d-1 classrooms, and they were all occupied. A new job arrives at time t, so the for some small ε > 0 gives a line crossing δ open interval.