**Lemma:** Given a scheduled set of jobs $A$ as decided by the greedy algorithm and a scheduled set of jobs $O$ as decided by an optimal algorithm, the following statement is true:

For every $i$th job in the set, $f(A_i) \leq f(O_i)$. That is, the finish time of the $i$th job in the greedy set is no later than the finish time of the corresponding job in the optimal set.

**Proof:**

**Base case:** For $i = 1$, the first job to be chosen by the algorithm, $f(A_1) \leq f(O_1)$ (because we know that the algorithm picks the job with the earliest deadline from the list).

**Hypothesis:** For some $i > 1$, assume that $f(A_i) \leq f(O_i)$.

**Induction:** After job number $i$ is added to both schedules, the greedy algorithm considers the set of compatible jobs, and then picks the job with the earliest finish time. We have two cases to consider here:

1. $f(A_i) = f(O_i)$, then all compatible jobs available to the optimal algorithm are also available to the greedy algorithm, and we guarantee the greedy algorithm will pick the job with earliest finish time. Then, $f(A_i) \leq f(O_i)$.

2. $f(A_i) < f(O_i)$, similarly all compatible jobs available to the optimal algorithm are also available to the greedy algorithm (the greedy algorithm might even have a larger set of compatible jobs to pick from), and we guarantee the greedy algorithm will pick the job with earliest finish time. Then, $f(A_i) \leq f(O_i)$.

**Theorem:** The greedy schedule is optimal.

In other words, we are trying to prove that $|A| = |O| = k$. We’ll do this by contradiction:

Assume that $|A| < k$, specifically (and without loss of generality) $|A| = k - 1$. From the Lemma above, we know that $f(A_{k-1}) \leq f(O_{k-1})$, which means that all compatible jobs available to the optimal algorithm, after the $k - 1$th job, are also available to the greedy algorithm. If the optimal schedule contains an extra job, then it’d also be in the greedy schedule. =¿ Contradiction