Given the greedy algorithm shown below,

\begin{verbatim}
EARLIEST-DEADLINE-FIRST (n, t_1, t_2, ..., t_n, d_1, d_2, ..., d_n)

SORT n jobs so that d_1 \leq d_2 \leq ... \leq d_n.

i \leftarrow 0

FOR j = 1 TO n

Assign job j to interval [t, t + t_j].

s_j \leftarrow t; f_j \leftarrow t + t_j

i \leftarrow i + t_j

RETURN intervals [s_1, f_1], [s_2, f_2], ..., [s_n, f_n].
\end{verbatim}

We’ll use the exchange approach to prove that the greedy algorithm produces an optimal schedule. The sequence of statements are as follows:

\textbf{Observation 1.} The earliest-deadline-first schedule has no idle time.

\textbf{Observation 2.} The earliest-deadline-first schedule has no inversions.

\textbf{Claim 1.} All schedules with no inversions and no idle time have the same maximum lateness.

\textbf{Claim 2.} There exists an optimal schedule with no idle time.

\textbf{Claim 3.} If a schedule (with no idle time) has an inversion, it has one with a pair of inverted jobs scheduled consecutively.

\textbf{Claim 4.} Swapping two adjacent, inverted jobs reduces the number of inversions by one and does not increase the max lateness.

\textbf{Theorem 1.} There is an optimal schedule that has no inversions and no idle time.

\textbf{Theorem 2.} The schedule \( A \) produced by the greedy algorithm has optimal maximum lateness \( L \).

Now let’s prove these claims and theorems.

\textbf{Claim 1.} All schedules with no inversions and no idle time have the same maximum lateness.

Consider two different schedules with with no inversions and no idle time for the same set of job. The difference between those two schedules would only be in the order of jobs with the same deadline (can you figure out why?). Among the jobs with that deadline \( d \), the last one has the maximum lateness, and this lateness does not depend on the order or size of the job.

\textbf{Claim 2.} There exists an optimal schedule with no idle time.

Think about this for a minute.

\textbf{Claim 3.} If a schedule (with no idle time) has an inversion, it has one with a pair of inverted jobs scheduled consecutively.

Consider an inversion in which a job \( a \) is scheduled sometime before another job \( b \), and \( d(a) > d(b) \). If we advance in the scheduled order of jobs from \( a \) to \( b \) one at a time, there has to come a point at which the deadline we see decreases for the first time. This corresponds to a pair of consecutive jobs that form an inversion.
Claim 4. Swapping two adjacent, inverted jobs reduces the number of inversions by one and does not increase the max lateness.
This is the hardest part of this proof.
Assume we have some optimal schedule \( O \) with maximum lateness of \( L \). Let \( \bar{O} \) indicate the new schedule we get after swapping jobs \( a \) and \( b \). It has a lateness of \( \bar{L} \).
After the swap, job \( b \) will have an earlier deadline, so we are not worried about it increasing the maximum lateness. We need to focus on job \( a \), which was pushed to a later deadline. Note that the new finish time of job \( a \) is now the old finish time of job \( b \).
Thus, the new lateness of \( a \) is \( \bar{l}(a) = f(a) - d(a) = f(b) - d(a) \).
Then, what can we say from this above equality?
Complete the following: \( \bar{l}(a) = f(b) - d(a) < \)

Theorem 1. There is an optimal schedule that has no inversions and no idle time. From claims 2, 3, and 4 above, we prove that any optimal schedule can be converted to a schedule that has no inversions and no idle time, without losing optimality.

Theorem 2. The schedule \( A \) produced by the greedy algorithm has optimal maximum lateness \( L \). From Theorem 1 and Claim 1, we prove that the schedule \( A \) will have the same maximum lateness as the optimal schedule.