The Greedy Scheduling Algorithm

Input: Tasks \( \{t_1, t_2, \ldots, t_n\} \), each with starting time \( s_i \) and finish time \( f_i \) (satisfying \( f_i \geq s_i \)).

Solution: Valid scheduling on \( m \) machines. Precisely, a function \( g: \{1, 2, \ldots, n\} \to \{1, 2, \ldots, m\} \) such that, for any \( 1 \leq i < j \leq n \), if \( g(i) = g(j) \), then tasks \( t_i \) and \( t_j \) do not overlap (that is, either \( f_i \leq s_j \) or \( f_j \leq s_i \)).

Measure: Minimize \( m \)

1 Sort the task according to starting time. At this moment, if \( i < j \), then \( s_i \leq s_j \).
2 \( m \leftarrow 0 \)
3 \textbf{for} \( i \leftarrow 1 \) \textbf{to} \( n \)
4 \hspace{1em} \textbf{if} (there is a machine \( j \) where task \( t_i \) fits)
5 \hspace{2em} \( g(i) \leftarrow j \) // put \( t_i \) on machine \( j \)
6 \hspace{1em} \textbf{else}
7 \hspace{2em} \( m \leftarrow m + 1 \)
8 \hspace{2em} \( g(i) \leftarrow m \) // put \( t_i \) on machine \( m \)
9 \hspace{1em} \textbf{endif}
10 \textbf{endfor}