p.422 16.1-2

This problem is identical to the original activity selection problem, but with inversed starting and ending times for any activity a_i.

Suppose $S_{optimal} = a_i$ for some $a_i \subseteq S$, where $a_i = [s_i, f_i)$ is the greedy solution. Reversing order, we can convert $a_i = [s_i, f_i)$ to $a'_i = [s'_i, f'_i)$, where $s'_i = f_i$ and $f'_i = s_i$. Inverting the s_i and f_i ensures that we select the first activity to finish. Hence, $S'_{optimal} = a'_i$ is the optimal greedy solution to the original problem.

p.447 16-2(a)

Greedy algorithm should suffice. Let Z_S denote an optimal sequence of tasks from the set S:

$$\{Z_S\} = \begin{cases} NUL & if S = \emptyset \\ \{a_i, Z_{S-\{a_i\}}\} & if S\emptyset \end{cases}, \, where \, p_i \leq p_j, \forall \, 1 \leq j \leq n \tag{1}$$

Let Z denote an optimal sequence of tasks given by the above recursion algorithm. We use contradiction to prove it is optimal. Suppose it is not optimal. That is, we schedule some a_j instead of a_i fist. Suppose this a_j is the k^{th} task in our original Z sequence. The average completion time is increased by $\frac{1}{n}(k-1)(p_j - p_i) > 0$. Thus, the original Z is the optimal sequence.

The running time would be the time complexity incurred from sorting the tasks by p_i. This can be done in $\theta(nlogn)$ using e.g. merge-sort.

p.637 23.2-8

We prove the correctness of the recursive scheme via induction.

Lets assume that the algorithm finds the MST T_1 for $G_1 = (V_1, E_1)$, and analogously for G_2. In the merge step, we are however not guaranteed to find a MST for the entire graph G. Suppose there exists $(v_i, v_k) \in T_1$, and $(v'_i, v'_k) \in T_2$ such that:

$$w(v_i, v_k) + w(v'_i, v'_k) > w(v_i, v'_i) + w(v_k, v'_k), \textit{ and}$$
$$\min w(v_i, v_k), w(v'_i, v'_k) > \min w(v_i, v'_i) + w(v_k, v'_k)$$

In the recursive algorithm, we would connect vertices (v_i, v_k), $w(v'_i, v'_k)$ and the light vertex, say (v_i, v'_i). Total weight incurred amounts to:

$$W = w(v_i, v_k) + w(v'_i, v'_k) + \min w(v_i, v'_i) + w(v_k, v'_k)$$

Should we connect vertices $(v_i, v'_i), (v_k, v'_k)$ and say (v_i, v_k), the lower total weight incurred amounts to:

$$W' = w(v_i, v'_i) + w(v_k, v'_k) + \min w(v_i, v'_i) + w(v_k, v'_k)$$

Hence $W > W'$.