(a) *Main Idea:* Use Selection Sort to rearrange the processes by non-decreasing time. But instead of actually reordering the elements, keep track of the index of each element as it is found.

```
MIN-COMPLETION(t_1, ..., t_n):
    T ← [t_1, ..., t_n]
    π ← []
    # Loop invariant: π contains the indices of the i − 1 smallest elements in T,
    # in non-decreasing order of the elements.
    for i ← 1, 2, ..., n:
        # Find the smallest element in T and store its index in π[i].
        π ← π + [1]   # start with element T[1]
        # Loop invariant: T[π[i]] is the smallest element in T[1 ... j − 1].
        for j ← 2, 3, ..., n:
            if T[j] < T[π[i]]:
                π[i] ← j
        # "Remove" process number π[i] from the input,
        # without changing the index of other processes.
        T[π[i]] ← ∞   # now T[π[i]] cannot be the smallest element again
    return π
```

(b) $\pi_0, \pi_1, \ldots, \pi_n$ are the *partial solutions* generated by the algorithm.

For the example given on the handout, $\pi_0 = []$, $\pi_1 = [1]$, $\pi_2 = [1,4]$, $\pi_3 = [1,4,3]$, $\pi_4 = [1,4,3,2]$.

(c) Partial solution π_k is *promising* iff some optimum solution π^* extends π_k (an optimum solution is any permutation of $[1,2,\ldots,n]$ that yields a minimum average completion time).

Optimum solution π^* *extends* partial solution π_k iff $\pi^*[i] = \pi_k[i]$ for $i = 1,2,\ldots,k$.

(d) We prove the loop invariant "π_k is promising" by induction on k (the number of iterations performed by the main loop—the one over variable i).

(e) There are **no** cases in the proof of the inductive step because the outcome of one iteration of the main loop is always the same: the smallest remaining element is found, its index is stored in $\pi[i]$, and it is changed to "∞" to remove it from consideration during future iterations.

(f) There are two "sub"-cases:

- if $\pi^*[k+1] = \pi_{k+1}[k+1]$;
- if $\pi^*[k+1] \neq \pi_{k+1}[k+1]$.

(g) Since every partial solution is promising, in particular, π_n (the value returned by the algorithm) is promising. This means some optimum permutation π^* extends π_n: $\pi^*[i] = \pi_n[i]$ for $i = 1,2,\ldots,n$. But then $\pi_n = \pi^*$, i.e., π_n itself is optimum.