Algorithm 3.1.57 Time-block optimization scheduler, by earliest time-block end times. Takes as input a list of 2-tuples in the form of $\langle start\ time, end\ time \rangle$.

```
1: procedure GREEDY SCHEDULER(a_1, a_2, ..., a_n: list of 2-tuples)
   ▷ Sort the list in nondecreasing order, by the 2nd position tuple elements.
2:     for j = 2 to n do
3:         i ← 1
4:         while a_{j_2} > a_{i_2} do
5:             i += 1
6:         end while
7:         element ← a_j
8:         for k = 0 to (j − i − 1) do
9:             a_{(j−k)} ← a_{(j−k−1)}
10:        end for
11:        a_i ← element
12:    end for                          ▷ List is sorted. End insertion sort.
13:    b ← [a_1]          ▷ a_1 has the least start time of all elements in a
14:    bindex ← 1
15:    for i = 2 to n do                ▷ Find the optimal schedule.
16:        if a_{i_1} ≥ b_{bindex_2} then
17:            bindex += 1
18:            b_{bindex} ← a_i
19:        end if
20:    end for
21:    return b_1, b_2, ..., b_m
22: end procedure
```