Assume indexes range from 1 to n

$$A = (a_1, a_2, \dots, a_n)$$

$$\forall a(a \in A \rightarrow a \in \mathbb{Z}_+)$$

(1) $leaders = \{a_i | a_i \in A\, ; (\forall j(i < j \rightarrow a_j < a_i) or\ i = n)\}$

(2) $(1) \rightarrow a_i = \max(A[i, i+1, \dots, n])$

From (1) it's logical to start iterate A from index n to index 1

The initial value of max is $A[\ n\ ]$.

At each iteration we compare the current element to max.

If we changed max then the current element will be added to $leaders$

$Leaders(A[\ n])$

1. $initialize\ List\ leaders = \emptyset$

2. $do{:}\ max \leftarrow A[n];\ leaders.addToHead(max)$

3. $for\ i \leftarrow n-1\ downto\ 1\ do$:

    3.1. $if\ max < A[i]\ then$ do:

        $max \leftarrow A[i]$

        $leaders.addToHead(max)$

Correctness:

(*) Loop Invariant:

$\forall 1 \le k \le n$

Before the $k'th$ iteration $max = \max(A[n-k+1,\ n-k+2, \dots, n])$

Base case:

$\big((1) + line\ 2\big) \rightarrow$before the first iteration $max = A[n-1+1] = A[n]$

Let $1 < k \le n$ assume (*) for k, therefore, before the $k'th$ iteration

$max = max(A[n-k+1, n-k+2, \dots, n])$

Observe the $k'th$ iteration:

We compare $A[n-k]$ to max and update max if needed.

In any case we get that after the $k'th$ iteration (before iteration $k+1$)

$max = \max(A[n-k, n-k+1, \dots, n]) = \max(A[n-(k+1)+1, n-(k+1)+2, \dots, n])$

As required.

Now, it's clear that if at some iteration we changed max then this element should be added to $leaders$ (at (3.1))