

Prove that Bubblesort

```
for i = 1 to n - 1
  for j = n downto i + 1
    if A[j] < A[j - 1]
      exchange A[j] with A[j - 1]
```

sorts. That is, prove that, with  $A' = \text{bubblesort}(A)$ ,

$$A'[1] \leq A'[2] \leq \dots \leq A'[n]$$

## Loop invariants

The proof will be based on loop invariants. These types of proofs have the following structure:

1. (Definition) Define loop invariant
2. (Initialization) Prove that it holds prior to the first iteration of the loop
3. (Maintenance) Prove that it holds after every iteration
4. (Termination) Prove that the loop terminates, and show that the loop invariant gives us a useful property for the overall proof.

## Invariant for the inner loop

### Definition

The inner loop has the invariant that the smallest value in the subarray  $A[i : n]$  is located in the (sub)subarray  $A[i : j]$ .

### Initialization

The invariant is true at the start of the loop since  $j = n$ , i.e.  $A[i : n] = A[i : j]$  - the smallest value is somewhere in the array.

### Maintenance

Let the index of the smallest value be  $k$ . We can have that  $k < j$  in which case the smallest value is still in the array  $A[i : j - 1]$  after the current iteration. Otherwise  $k = j$  (by assumption the smallest value of  $A[i : n]$  must be in  $A[i : j]$ , i.e.  $i \leq k \leq j$ ). If this is the case, then the **if**-statement will be true, and the smallest value is swapped and placed at  $j - 1$ .

Therefore, the loop invariant is true after an iteration if it were true before.

### Termination

When the inner loop terminates, that is  $j = i$  (we are placed *after* the execution of the inner loop), the smallest value of  $A[i : n]$  is contained in the subarray  $A[i : i]$ . That is, the smallest value of  $A[i : n]$  has been placed at the start of the array, at index  $i$ .

## Invariant for the outer loop

### Definition

The subarray  $A[1 : i - 1]$  is sorted and  $A[p] \leq A[q]$  for all  $p \in [1 : i - 1]$  and  $q \in [i : n]$ .

### Initialization

The invariant is true at the start where  $i = 1$  if we define  $A[1 : 0]$  to be the empty array.

### Maintenance

By the inner loop invariant we have that *after* the execution of the loop body the smallest value in  $A[i : n]$  is at index  $i$ . By the outer loop invariant we then have that  $A[1 : i]$  is sorted and  $A[p] \leq A[q]$  for all  $p \in [1 : i]$  and  $q \in [i + 1 : n]$ .

### Termination

The outer loop terminates when  $i = n$ , so we have that the array is sorted when the algorithm terminates.

## Running time

At step  $i$  the inner loop executes  $n - i$  steps, therefore the total number of steps is

$$n_{steps} = \sum_{i=1}^{n-1} (n - i) = n(n - 1) - (n)(n - 1)/2 = \frac{n(n - 1)}{2}$$

Bubblesort runs in  $O(n^2)$  time.