

Name Muhammad Larai'b Akhter  
Section BCS-4B  
Course Design and Analysis of Algorithms.  
Roll no. 211-5294

a) Find min  
Minimum(A)  
1 min = A[1]  
2 for i = 2 to A.length  
3 if min > A[i]  
4 min = A[i]  
5 return min

Loop Invariant: At the start of each iteration 'min' holds the value of smallest element A[1...A.length]

Initialization: Before first iteration 'min' is initialized to A[1] so loop invariant holds first iteration.

Maintenance: At the end of each iteration, if  $A[i] < \text{min}$  then min is updated with the value of A[i]. Min then holds the smallest element in A[1...i].

Termination: Loop terminates when i is equal to A.length so min holds the smallest element in A[1...A.length].

b) BubbleSort(A)

- 1 for  $i = 1$  to  $A.length - 1$
- 2     for  $j = A.length$  down to  $i + 1$
- 3         if  $A[j] < A[j - 1]$
- 4             exchange  $A[j]$  with  $A[j - 1]$

loop invariant: At start of each iteration of outer loop elements in  $A[1 \dots i]$  are sorted.

Initialization: Before the first iteration  $i = 1$  so  $A[1 \dots 0]$  is empty. Therefore loop invariant holds.

Maintenance: At the end of each outer loop iteration, largest element in  $A[i \dots A.length]$  bubbles up to the end and  $A[1 \dots i]$  is still sorted.

Termination: loop terminates when  $i = A.length - 1$  hence  $A[1 \dots A.length - 1]$  is sorted since inner loop insures the largest element is at the end of each subarray.



### C) SelectionSort(A)

```
1 for j=1 to A.length-1
2   smallest_index = j
3   for i=j+1 to A.length
4     if A[i] < A[smallest_index]
5       smallest_index = i
6   exchange A[j] with A[smallest_index]
```

loop invariant: At the start of each iteration of outer loop elements in  $A[1 \dots j-1]$  are sorted.

Initialization: Before first iteration  $j=1$  so  $A[1 \dots 0]$  is empty. therefore first iteration holds.

Maintenance: At the end of each iteration of outer loop ~~min~~ smallest\_index stores the index of smallest element in  $A[i \dots A.length]$ . Then the  $A[\text{smallest\_index}]$  is swapped with  $A[j]$ . Hence array is sorted  $A[1 \dots j-1]$ .

Termination: When outer loop terminates  $j == A.length-1$  and  $A[j \dots A.length-1]$  is sorted, Since smallest\_index gets the element of the smallest index in  $A[1 \dots A.length]$  and ~~put in~~ <sup>swaps with</sup>  $A[j]$ . Hence A is sorted.