| | |
|---|---|
| Name | Muhammad Laraib Akhtar |
| Section | BCS-4B |
| Course | Design and Analysis of Algorithms. |
| Roll no. | 21L-5294 |

a) Find min

```
Minimum(A)
min = A[1]
for i = 2 to A.length
    if min > A[i]
        min = A[i]
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 invaiant holds first iteration.

Maintenance: At the end of each iteration, if A[i] < 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)
for i=1 to A.length-1
    for j= A.length downto i+1
        if A[j] < A[j-1]
            exchange A[j] with A[j-1]
```

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

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

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

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

c) SelectionSort(A)

```
for j=1 to A.length-1
    smallest_index = j
    for i=j+1 to A.length
        if A[i]<A[smallest_index]
            smallest_index=i
    exchange A[j] with A[smallest_index
```

loopinvariant: At the start of each iteration of outerloop elements in A[1 ... ~~A~~j-1] are sorted.

Initialization: Before first iteration j==1 so A[1.... 0] is emptly. therefor first iteration holds.

Maintenance: At the end of each iteration of outerloop ~~[illegible]~~ smallest_index stores the index of smallest element in A[i.... A.length]. ~~At~~ Then the A[smallest_index] is swapped with A[j]. Hence array is sorted A[~~g~~ 1.... j-1].

Termination: when outer loop terminates j == A.length-1 and A[j.... A.length-1] is sorted, Since smallest_index gets the element of the smallest index in A[1.... A.length] and ~~put in~~ swaps with A[i]. Hence A is sorted.