

Rohith Janwaalkar

1002226469

Correctness of Selection Sort Algorithm

Find the smallest element in the unsorted part of array and swap it with the first element in that part of array.

```
For i=0 to n-1 {  
    minimum = i  
    for j=i+1 to n {  
        if (arr[j] < arr[min]) {  
            minimum = j  
        }  
    }  
    if (i != min) {  
        swap (ith element, minimum element)  
    }  
}
```

Correctness: The correctness of the Selection Sort algorithm can be proven by using a loop invariant and demonstrating that it holds true during the execution of algorithm.

Initialization:

Invariant

At the start of each iteration of the outer loop, the subarray before the current index 'i' is sorted, and all elements in this subarray are smaller than or equal to any element in the unsorted portion of array.

for $i=0$ to $n-1$ {

 minimum = i

 3

Maintainance :- Before the execution of inner loop, all the elements to before i are sorted. The inner loop identifies the minimum element in the array and swaps the values accordingly.

for ($j=i+1$ to n) {

 if ($arr[j] < arr[minum]$) {

 minimum = j

 3

 3

 swap ($arr[i]$, $arr[minum]$)

The algorithm sorts the element at index P which belongs at the index $minum$.

The outer loop runs from 0 to $n-1$, iterating through array.

Termination :- Once the outer loop completes, the algorithm terminates. The entire array is sorted, and there are no infinite loops of conditions that would prevent the completion.