

## **\*\*SELECTION SORT CORRECTNESS\*\***

```
def selection_sort(arr):  
    for i in range(len(arr)):  
        min_idx = i  
        for j in range(i + 1, len(arr)):  
            if arr[j] < arr[min_idx]:  
                min_idx = j  
        arr[i], arr[min_idx] = arr[min_idx], arr[i]
```

loop invariant: At the start of each iteration of the first loop for each  $i$  the sub array goes from 0 to  $i-1$ , which is already sorted. The smallest element from the array which is currently pointed by  $j$  (of inner loop) of the unsorted array portion will be placed in the  $i$ th position by the end of the iteration.

### Initialization:

Before the first iteration no element is sorted as the outer loop starts with  $i = 0$  so the invariant

EX: arr[5,10,2,4,6]

$i=0$

min value is 2 so it is replaced with  $i$

arr[2,10,5,4,6]

### Maintenance:

After the  $i$ -th iteration the smallest element in the unsorted array is correctly placed at the  $i$ -th position, (Unsorted array =  $i$  to  $n-1$ )

Now the array is divided into two sub arrays where 0 to  $i$  is now sorted and  $i+1$  to  $n-1$  is unsorted.

EX:  $i = 1$

current unsorted sub array arr[10,5,4,6]

after swapping arr[4,5,10,6]

sorted = arr[2,4]      unsorted = [5,10,6]

sorted = arr[0... $i$ ]      unsorted = arr[ $i$ .... $n-1$ ]

\* loop variance holds

Termination:

When  $i = n-1$  (Final iteration), the entire array is already sorted and each element is correctly placed.

EX:  $i = 4$

sorted = arr[2,4,5,6] unsorted = arr[10]

here  $i=n-1$

so the complete array = [2,4,5,6,10]

Thus array is coorectly sorted by the end.