

```
static void selectionSort(int arr[], int n){
    int i, j, min_ind, temp;
    for(i=0; i<=n-2; i++){
        min_ind = i;
        for(j=i+1; j<=n-1; j++){
            if(arr[min_ind] > arr[j]){
                min_ind = j;
            }
        }
        temp = arr[min_ind];
        arr[min_ind] = arr[i];
        arr[i] = temp;
    }
}
```

$j=n$  (out of boundary!!)

TC  $\rightarrow O(n \times n) = O(n^2)$

SC  $\rightarrow O(1)$

Time  $O(2n)$

```
for(i=0; i<=n; i++) ← (n+1)
{
    for(j=1; j<=2; j++) → (n+1) × 2
    {
        print("Hello");
    }
}
```

TC =  $2n+2 = O(n)$

| i | i <= N   |
|---|----------|
| 0 | 0 <= 4 ✓ |
| 1 | 1 <= 4 ✓ |
| 2 | 2 <= 4 ✓ |
| 3 | 3 <= 4 ✓ |
| 4 | 4 <= 4 ✓ |
| 5 | 5 <= 4 ✗ |

Select minimum.

| j | j <= 2   |
|---|----------|
| 1 | 1 <= 2 ✓ |
| 2 | 2 <= 2 ✓ |
| 3 | 3 <= 2 ✗ |

min\_ind = i = 0

j = 1: arr[min\_ind] < arr[j] → min\_ind = 1

j = 2: arr[min\_ind] > arr[j] → min\_ind = 2

j = 3: arr[min\_ind] > arr[j] → min\_ind = 3

j = 4: arr[min\_ind] > arr[j] → min\_ind = 4

Swap(arr[min\_ind], arr[i])

loop-2  $\rightarrow i, j$

Selection sort

14 15 13 12 11

i = 0

j = i + 1

min\_ind = 1

11 15 13 12 14

11 15 13 12 14  
sorted unsorted

$i=1$   
 $j=i+1$

11 12 13 15 14  
sorted unsorted

$i=2$   
11 12 13 15 14  
sorted

$j=i+1$

11 12 13 15 14  
sorted unsorted

$i=3$   
 $j=i+1$

swap (arr[...], arr[...])

$\min\_ind = i = 1 \neq 3$   
 $j=2, \text{arr}[\min\_ind] > \text{arr}[2]$   
 $j=3, \text{arr}[\min\_ind] > \text{arr}[3]$   
 $j=4, \text{arr}[\min\_ind] < \text{arr}[4]$   
 swap (arr[min\_ind=3], arr[i=1])

$\min\_ind = 2$   
 $j=3, \text{arr}[\min\_ind] < \text{arr}[3]$   
 $j=4, \text{arr}[\min\_ind] < \text{arr}[4]$   
 swap (arr[min\_ind=2], arr[i=2])

$\min\_ind = 3 \neq 4$   
 $j=4, \text{arr}[\min\_ind] > \text{arr}[4]$   
 $j=5, \text{Out of Boundary}$

$j = 1$

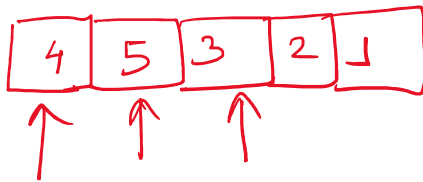
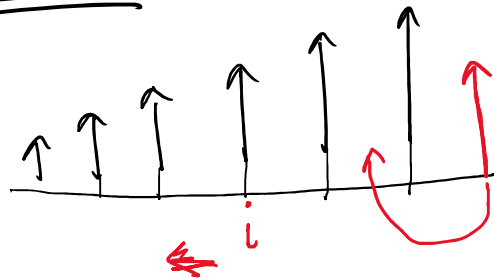
11 12 13 14 15

$j = 5$ , Out of Boundary  
 $\text{swap}(\text{arr}(\text{min-ind} = 4), \text{arr}(i = 3))$

$i = 4$

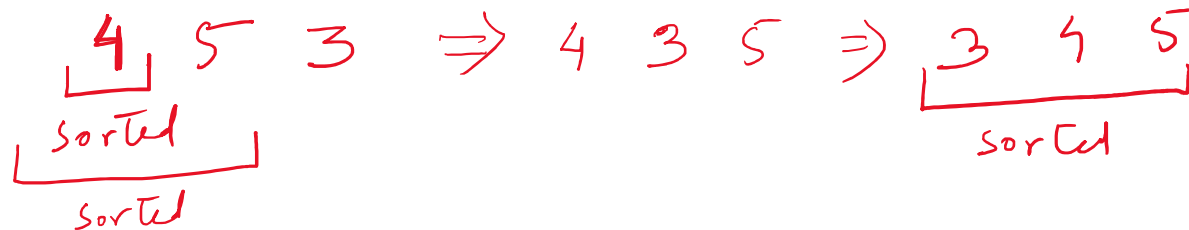
Ex:  $n = 5$ ,  $i = 3 = n - 2$

Insertion Sort



Note on insertion sort -

1st element  $\rightarrow$  Sorted.



3 4 5 2  $\Rightarrow$  3 4 2 5  $\Rightarrow$  3 2 4 5  $\Rightarrow$  2 3 4 5

2 3 4 5 1  $\Rightarrow$  1 2 3 4 5

static void insertionsort(int arr[], int n)  
{  
for (int i = 1; i < n; ++i) {  
int key = arr[i];  
int j = i - 1;  
while (j >= 0 && arr[j] > key) {  
arr[j + 1] = arr[j];  
j = j - 1;  
}  
arr[j + 1] = key;  
}  
}

n times  
n times

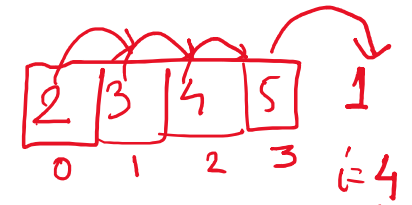
TC =  $O(n^2)$   
SC =  $O(1)$

0  
sorted

arr[4+1] = key j = -1  
arr[0] = 1

ind = 0, 1, 2, 3, ..., i-3, i-2, i-1, i

key = 1



key = arr[i] = 1

j = i - 1 = 4 - 1 = 3

arr[3] = 5 > key = 1

arr[3+1] = arr[4] = arr[3]

arr[4] = 5