

# Sorting

## Selection Sort :-

Key words :- Select min

min = 0 | min = num = 5  
temp = 9 | arr[1] →

n-2  
0, 1, 2, 3, 4

\* Get the min and swap it

0	1	2	3	4	5
13	46	24	52	20	9

13, 46, 24, 52, 20, 9  
0 1 2 3 4 5

↓  
9, 13, 20, 24, 46, 52

9 | 46, 24, 52, 20, 13  
↓  
Sorted | Unsorted

Step 1

9, 13 | 24, 52, 20, 46  
Sorted | Unsorted

Step 2

9, 13, 20 | 52, 24, 46  
Sorted | Unsorted

Step 3

n-1 steps

9, 13, 20, 24 | 52, 46  
Sorted | Unsorted

Step 4

9, 13, 20, 24, 46, 52  
Sorted

Step 5

## Pseudo Code

Swap at node 0 { min(array) { 0 - n-1 } }

Swap at node 1 { min(array) { 1 - n-1 } }

⋮  
n-2

```
for (i=0; i <= n-2; i++)
```

```
{   min = i
```

```
    for (j=i; j <= n-1; j++)
```

```
    {   if (arr[j] < arr[min])
```

```
        {   min = j
```

```
    }
```

```
    arr[min] = arr[i]
```

```
    temp = arr[min]
```

```
    arr[min] = arr[i]
```

```
    arr[i] = temp
```

time complexity =  $O(n^2)$

### \* Key points to remember

→ Assume the first element in an array is min and update the value

→ Find the min and swap it to the first element.