

Selection Sort

link -

https://www.youtube.com/watch?v=dQa4A2Z0_Ro&t=141s

Define

Find the minimum element in unsorted array and swap it with element at beginning.

Example -

input

5

5 4 3 2 1

output

1 2 3 4 5

code

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    long long int n,i;
    cin>>n;
    int arr[n];
    for(i=0;i<n;i++)
    {
        cin>>arr[i];
    }

    for (i=0;i<n-1;i++)
    {
        for(int j =i+1;j<n;j++)
        {
            if(arr[j]<arr[i])
            {
                int temp = arr[j];
                arr[j] = arr[i];
                arr[i] = temp;
            }
        }
    }
}
```

```
}
```

```
}
```

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

```
{
```

```
    cout<<arr[i]<<" ";
```

```
}
```

```
cout<<endl;
```

```
return 0;
```

```
}
```

Bubble Short

link -

<https://www.youtube.com/watch?v=xcPFUCh0jT0&t=319s>

Define -

Repeatedly Swap two adjacent elements if they are in wrong order

Example-

input

5

6 3 8 2 4

Output

2 3 4 6 8

code

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    long long int n,i;
    cin>>n;

    int arr[n];

    for(i=0;i<n;i++)
    {
        cin>>arr[i];
    }

    int counter = 1;
    while(counter < n)
    {
        for(int i=0;i< n - counter;i++)
        {
            if(arr[i] > arr[i+1])
            {
                int temp = arr[i];
```

```
arr[i] = arr[i+1];  
arr[i+1] = temp;
```

```
}
```

```
}
```

```
counter++;
```

```
}
```

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

```
{
```

```
    cout<<arr[i]<<" ";
```

```
}
```

```
cout<<endl;
```

```
return 0;
```

```
}
```

Sorting (8.3)

1. Selection Sort:

Idea: The inner loop selects the minimum element in the unsorted array and places the elements in increasing order.

Time complexity: $O(N^2)$

```
for (int i = 0; i < n - 1; i++) {  
    for (int j = i + 1; j < n; j++) {  
        if (arr[j] < arr[i]) {  
            int temp = arr[j];  
            arr[j] = arr[i];  
            arr[i] = temp;  
        }  
    }  
}
```

2. Bubble Sort:

Idea: if $\text{arr}[i] > \text{arr}[i+1]$ swap them. To place the element in their respective position, we have to do the following operation $N-1$ times.

Time Complexity: $O(N^2)$

```
int counter = 0;  
while (counter < n - 1) {  
    for (int i = 0; i < n - counter - 1; i++) {  
        if (arr[i] > arr[i + 1]) {  
            int temp = arr[i];  
            arr[i] = arr[i + 1];  
            arr[i + 1] = temp;  
        }  
    }  
    counter++;  
}
```

3. Insertion Sort:

Idea: Take an element from the unsorted array, place it in its corresponding position in the sorted part, and shift the elements accordingly.

Time Complexity: $O(N^2)$

```
for (int i = 1; i < n; i++) {  
    int current = arr[i];  
    int j = i - 1;  
    while (arr[j] > current && j >= 0) {  
        arr[j + 1] = arr[j];  
        j--;  
    }  
    arr[j + 1] = current;  
}
```

Homework: Implement selection sort, bubble sort, insertion sort on your own.