

Code:

```
import java.util.*;

public class Main
{
    public static void main(String[] args) {
        Scanner s=new Scanner(System.in);

        System.out.println("Enter the number of elements");
        int n=s.nextInt();
        int[] arr=new int[n];
        System.out.println("Enter the elements");
        for(int i=0;i<n;i++)
        {
            arr[i]=s.nextInt();
        }
        for(int i=0;i<n-1;i++)
        {
            for(int j=i+1;j<n;j++)
            {
                if(arr[i]>arr[j])
                {
                    int temp=arr[i];
                    arr[i]=arr[j];
                    arr[j]=temp;
                }
            }
        }
        System.out.println("After sorting:");
        for(int i=0;i<n;i++)
        {
            System.out.print(arr[i]+" ");
        }
    }
}
```

} }

Output:

```
Enter the number of elements
5
Enter the elements
15 13 21 11 10
After sorting:
10 11 13 15 21

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Enter the number of elements
10
Enter the elements
3 6 2 4 8 7 9 1 5 10
After sorting:
1 2 3 4 5 6 7 8 9 10

...Program finished with exit code 0
Press ENTER to exit console.
```

Time complexity and space complexity:

$$\begin{aligned}
 & \text{for } (i = 0; i < \text{Size} - 1; i++) \\
 & \quad \text{for } (j = i + 1; j < \text{Size}; j++) \\
 & \quad \quad \text{if } (arr[i] > arr[j]) \\
 & \quad \quad \quad \text{swap}(arr[i], arr[j])
 \end{aligned}$$

$$\begin{aligned}
 & \text{Size} - 1 \\
 & \quad 1 \quad 2 \quad \frac{n(n-1)}{2} \quad \text{swap}(arr[i], arr[j]) \\
 & \quad 3 \\
 & \quad 4 \quad 3 \\
 & \quad \vdots \\
 & \quad \text{Size} - 1 \\
 & \quad \vdots \\
 & \quad n - 1 \quad \frac{n(n-1)}{2} \\
 & \quad \quad 1 \quad \frac{n^2 - n + 2n - 2}{2} \\
 & \quad \quad \frac{n(n-1)}{2} \quad \frac{n^2 - n - 2}{2} \\
 & \quad \quad \quad 2
 \end{aligned}$$

Ignoring lower order exponents
 Time complexity $\Rightarrow O(n^2)$

Space complexity

arr[] \rightarrow of n words

$i \rightarrow$ 1 word

$j \rightarrow$ 1 word

size \rightarrow 1 word

$$f(n) = n + 3$$

ignoring exponent values and coefficients,
space complexity = $O(n)$