

<https://course.acciojob.com/idle?question=023eeb5a-1016-4aba-a007-fbfc577f00a3>

- EASY
- Max Score: 30 Points

Running Time

The running time of an algorithm for a specific input depends on the number of operations executed. The greater the number of operations, the longer the running time of an algorithm. We usually want to know how many operations an algorithm will execute in proportion to the size of its input, which we will call N .

In case of Insertion Sort, For each element v in an array of N numbers, Insertion Sort compares the number to those to its left until it reaches a lower value element or the start. At that point, it shifts everything to the right-up one and inserts v into the array.

How long does all that shifting take? In the best case, where the array was already sorted, no element will need to be moved, so the algorithm will just run through the array once and return the sorted array. The running time would be directly proportional to the size of the input, so we can say it will take N time. However, we usually focus on the worst-case running time (computer scientists are pretty pessimistic). The worst case for Insertion Sort occurs when the array is in reverse order. To insert each number, the algorithm will have to shift over that number to the beginning of the array. Sorting the entire array of N numbers will therefore take $1+2+3+\dots+(N-1)$ operations, which is $N(N-1)/2$ (almost $(N^2)/2$). Computer scientists just round that up (pick the dominant term) to (N^2) and say that Insertion Sort is an " (N^2) time" algorithm.

The only thing we should print is the number of shifts made by the algorithm to completely sort the array. A shift occurs when an element's position changes in the array. Do not shift an element if it is not necessary.

Input Format

The first line inputs N , the size of the integer array.

The second line inputs N space-separated integers, i.e., $A[i]$.

Output Format

Print the number of shifts it will take to sort the array

Example 1

Input:

```
5
2 1 3 1 2
```

Output:

```
4
```

Explanation

Iteration	Array	Shifts
0	2 1 3 1 2	0
1	1 2 3 1 2	1
2	1 2 3 1 2	0
3	1 1 2 3 2	2
4	1 1 2 2 3	1
Total		4

Example 2

Input:

```
5
2 1 3 1 3
```

Output:

```
3
```

Explanation:

3 shifts will be required

Constraints

$1 \leq N \leq 1001$

$-10000 \leq A[i] \leq 10000$

Topic Tags

- [Sorting](#)
- [Arrays](#)

My code

// in java

```
import java.util.*;  
import java.lang.*;  
import java.io.*;
```

```
public class Main  
{
```

```
    public static void main (String[] args) throws java.lang.Exception  
    {
```

```
        //your code here
```

```
        Scanner s=new Scanner(System.in);
```

```
        int n=s.nextInt();
```

```
        int arr[]=new int[n];
```

```
        int c=0;
```

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

```
            arr[i]=s.nextInt();
```

```
        for(int i=0;i<n-1;i++)
```

```
            if(arr[i]>arr[i+1])
```

```
{ int j=i;
  while(j>-1 && arr[j]>arr[j+1] )
  {
    c=c+1;
    int t=arr[j];
    arr[j]=arr[j+1];
    arr[j+1]=t;
    j--;
  }
}
// for(int i=0;i<n;i++)
System.out.print(c);
}
```

```
}
```