All Tracks > Algorithms > Sorting > Insertion Sort

Algorithms

Solve any problem to achieve a rank

View Leaderboard

Topics: Insertion Sort

Insertion Sort

TUTORIAL PROBLEMS VISUALIZER BETA

Insertion sort is based on the idea that one element from the input elements is consumed in each iteration to find its correct position i.e, the position to which it belongs in a sorted array.

It iterates the input elements by growing the sorted array at each iteration. It compares the current element with the largest value in the sorted array. If the current element is greater, then it leaves the element in its place and moves on to the next element else it finds its correct position in the sorted array and moves it to that position. This is done by shifting all the elements, which are larger than the current element, in the sorted array to one position ahead

Implementation

```
void insertion_sort ( int A[ ] , int n)
{
     for( int i = 0 ;i < n ; i++ ) {
    /*storing current element whose left side is checked for its
             correct position .*/

      int temp = A[ i ];
      int j = i;

       /* check whether the adjacent element in left side is greater or
```

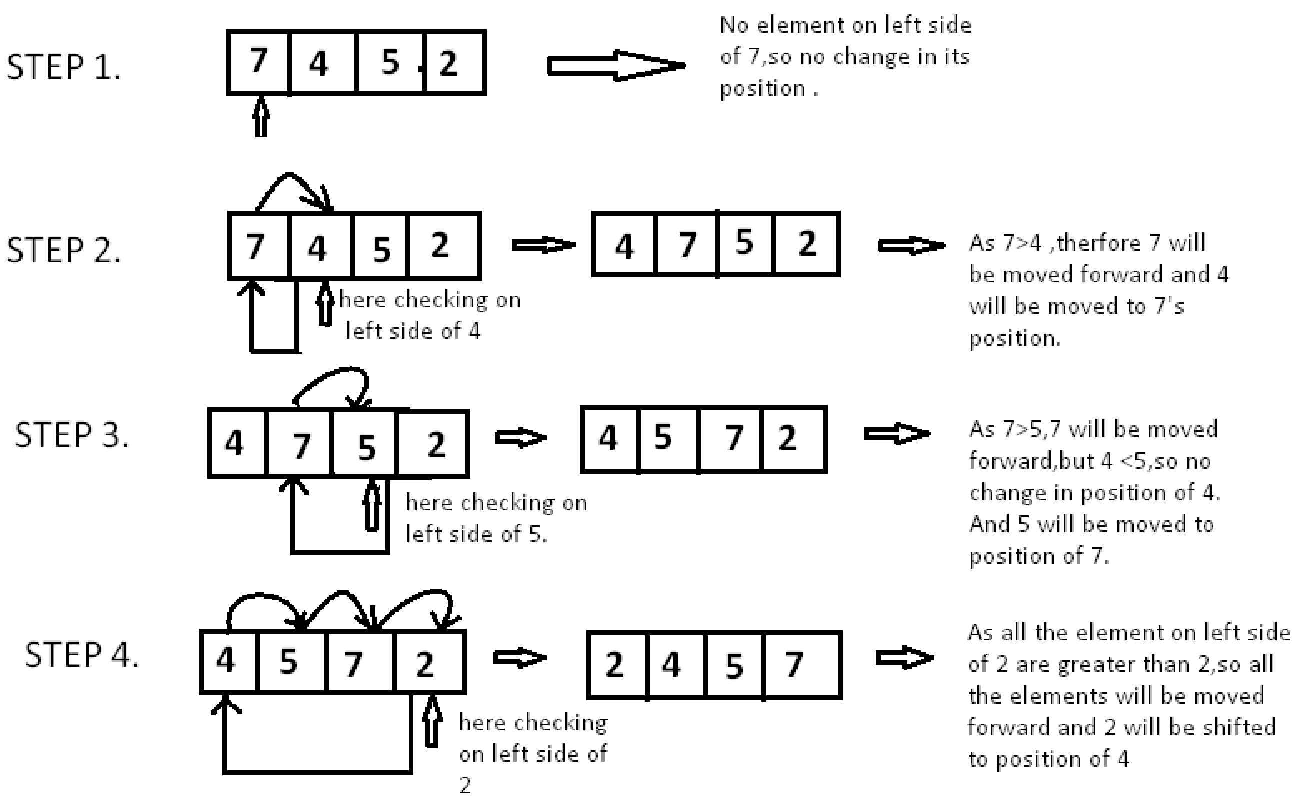
```
              less than the current element. */

            while(  j > 0  && temp < A[ j -1]) {

             // moving the left side element to one position forward.
                  A[ j ] = A[ j-1];
                  j= j - 1;

             }
           // moving current element to its  correct position.
             A[ j ] = temp;
       }
}
```

Take array $A[] = [7, 4, 5, 2]$.


STEP 1.
7 4 5 2
No element on left side of 7,so no change in its position .
STEP 2.
7 4 5 2
here checking on left side of 4
4 7 5 2
As 7>4 ,therfore 7 will be moved forward and 4 will be moved to 7's position.
STEP 3.
4 7 5 2
here checking on left side of 5.
4 5 7 2
As 7>5,7 will be moved forward,but 4 <5,so no change in position of 4. And 5 will be moved to position of 7.
STEP 4.
4 5 7 2
here checking on left side of 2
2 4 5 7
As all the element on left side of 2 are greater than 2,so all the elements will be moved forward and 2 will be shifted to position of 4

Since 7 is the first element has no other element to be compared with, it remains at its position. Now when on moving towards 4, 7 is the largest element in the sorted list and greater than 4. So, move 4 to its correct position i.e. before 7. Similarly with 5, as 7 (largest element in the sorted list) is greater than 5, we will move 5 to its correct position. Finally for 2, all the elements on the left side of 2 (sorted list) are moved one position forward as all are greater than 2 and then 2 is placed in the first position. Finally, the given array will result in a sorted array.

Time Complexity:

In worst case,each element is compared with all the other elements in the sorted array. For N elements, there will be N^2 comparisons. Therefore, the time complexity is $O(N^2)$

Contributed by: Anand Jaisingh

Did you find this tutorial helpful? YES NO

TEST YOUR UNDERSTANDING

Insertion Sort

You have been given an *A* array consisting of *N* integers. All the elements in this array are guaranteed to be unique. For each position *i* in the array *A* you need to find the position $A[i]$ should be present in, if the array was a sorted array. You need to find this for each *i* and print the resulting solution.

Input Format:

The first line contains a single integer *N* denoting the size of array *A*. The next line contains *N* space separated integers denoting the elements of array *A*.

Output Format:

Print *N* space separated integers on a single line , where the *i*th integer denotes the position of $A[i]$ if this array were sorted.

Constraints:

$$1 \leq N \leq 100$$

$$1 \leq A[i] \leq 100$$

SAMPLE INPUT

```
5
1 2 3 4 5
```

SAMPLE OUTPUT

```
1 2 3 4 5
```

Enter your code or Upload your code as file. Save C (gcc 5.4.0) ▾

```
/*
// Sample code to perform I/O:
#include <stdio.h>

int main(){
    int num;
    scanf("%d", &num);                          // Reading input from STDIN
    printf("Input number is %d.\n", num);       // Writing output to STDOUT
}

// Warning: Printing unwanted or ill-formatted data to output will cause the test cases t
*/

// Write your code here

```

1:1

Provide custom input

COMPILE & TEST SUBMIT

COMMENTS (78)

SORT BY: Relevance▾

Login/Signup to Comment

tsolerau Dec 22, 2016 at 12:17 PM

can someone explain me what should be output? I don't quite understand this For each position i in the array A you need to find the position A[i] should be present in, if the array was a sorted array. You need to find this for each i and print the resulting solution.

▲ 3 votes ● Reply ● Message ● Permalink

Ajish Athrayil Jan 29, 2017 at 02:52 AM

You need to print out the position of each element of the unsorted array that it'll take after the array is sorted .
for eg:-
input
5
9 7 8 12 10

7 8 9 10 12 -- sorted array

output-- 3 1 2 5 4

▲ 37 votes ● Reply ● Message ● Permalink

Sreerag Nair Jun 11, 2017 at 06:38 AM

Thanks a lot, man!

▲ 0 votes ● Reply ● Message ● Permalink

Ajish Athrayil Jun 12, 2017 at 06:48 AM

Glad it helped.

▲ 0 votes ● Reply ● Message ● Permalink

Chanukya Nunna Nov 12, 2016 at 06:21 AM

In the above insertion sort algorithm it should be while(j > 0 &&temp < A[j -1]) not while(temp < A[j -1] && j > 0)

▲ 10 votes ● Reply ● Message ● Permalink

Prateek Garg Nov 18, 2016 at 05:49 AM

Thanks for pointing it out. It has been fixed.

▲ 0 votes ● Reply ● Message ● Permalink

Ajith kumar Nov 24, 2016 at 12:33 AM

what is the difference between both the statements?

▲ 2 votes ● Reply ● Message ● Permalink

Kumar Yashwant Nov 26, 2016 at 11:38 PM

I think it will save one spare comparison as well as will avoid comparison with non-existent array elements.
When j = 0, 1st will pass, the second will fail.

▲ 4 votes ● Reply ● Message ● Permalink

Akash Sharma ϟ Admin Dec 23, 2016 at 03:43 AM

If j = 0, first checking temp < A[j-1] will throw runtime exception. So its better to first check if j > 0 and then comparing temp and A[j-1].

▲ 13 votes ● Reply ● Permalink

Hemant Mangwani Sep 27, 2017 at 01:56 AM

this is because && and gate if 1 condition is false it will not check the second condition

▲ 0 votes ● Reply ● Message ● Permalink

Dinesh ✎ Edited May 13, 2018 at 12:13 PM

&& is short-circuit operator, it means if 1st condition is failed then there is no use of checking other condition since it will be anyway false. Main reason: www.hackerearth.com/practice/algorithms/sorting/insertion-sort/tutorial/?scroll-id=comments-563-41&scroll-trigger=inview#c78907

▲ 0 votes ● Reply ● Message ● Permalink

Mustapha Sahli Nov 25, 2016 at 09:01 AM

there is a problem in the second and third inputs the correct answers are not correct

▲ 5 votes ● Reply ● Message ● Permalink

Nhan Ly Dec 29, 2016 at 01:52 AM

I have the same issue. Admin please check it

▲ 1 vote ● Reply ● Message ● Permalink

Peter Pan Jan 02, 2017 at 07:03 AM

I second that.

▲ 1 vote ● Reply ● Message ● Permalink

Andrew Case Jan 20, 2017 at 12:52 AM

I can confirm that both the second and third "correct" answers are unsorted arrays.

▲ 2 votes ● Reply ● Message ● Permalink

Guilherme Rodrigues Feb 17, 2017 at 09:32 AM

Same issue. Just confirmed with excel that at least the 2nd "correct" answers does not match a sorted array of indices to the original array.

▲ 0 votes ● Reply ● Message ● Permalink

Sreerag Nair Jun 11, 2017 at 06:45 AM

Look above for Ajish Athrayil's explanation. Try solving it that way and u will get the answer right.
What u basically have to do is this :
1. sort the array.
2. get the position of each element in the sorted array.
3. print the position of the element as per the question.

eg.
3
7 2 3
sorted output : 2 3 7
position : 1 2 3
output : 3 1 2

▲ 0 votes ● Reply ● Message ● Permalink

Jenish Patel Apr 14, 2017 at 04:08 AM

No it's Right
I got all correct.

▲ 3 votes ● Reply ● Message ● Permalink

Vijitha Gunta Dec 22, 2016 at 05:27 AM

Isn't it easier to solve this with selection sort than with insertion sort?

▲ 3 votes ● Reply ● Message ● Permalink

Aman Jain Sep 18, 2017 at 04:10 PM

well you can use inbuilt sort function too,but this problem here to practise insertion sort

▲ 5 votes ● Reply ● Message ● Permalink

RS156 Edited Jan 30, 2018 at 11:00 PM

When using inbuilt sort function the question will become unnecessarily tedious as all the original indices will be lost. Comparing the original array with the sorted array will be much more bothersome rather than creating a separate array counting the index change.
It will be easier to solve this question using selection sort rather than insertion as the index is directly stored in minimum variable in each iteration which can be directly printed as the answer. So selection sort will require minimum changes in the code.

▲ 0 votes ● Reply ● Message ● Permalink

Nikhil Kumar Feb 21, 2017 at 07:20 AM

This can be silly question...but Whats the point of insertion sort here? I can simply use Arrays.sort (Sort it with any method) and then find index accordingly. Can anyone explain how insertion sort can be useful here?

▲ 1 vote ● Reply ● Message ● Permalink

Akshay S Danthi Edited Mar 12, 2017 at 12:03 PM

Same. Did you find any reason later?

▲ 0 votes ● Reply ● Message ● Permalink

N Nair Apr 20, 2017 at 12:36 PM

Isn't the idea just to apply insertion sort? This is a question to supplement the tutorial.

▲ 1 vote ● Reply ● Message ● Permalink

Nikhil Kumar Apr 20, 2017 at 12:40 PM

May be.... by the way check out, tutorial question for merge sort... its really good

▲ 0 votes ● Reply ● Message ● Permalink

N Nair Apr 20, 2017 at 01:26 PM

Sure I will. Thanks :)

▲ 0 votes ● Reply ● Message ● Permalink

Terry Gruenewald Apr 01, 2017 at 11:26 PM

For Java8, all the correct output is wrong. For example, input #1 is "100 99 98 97 96 95 94 .... 4 3 2 1" and the correct output says it should be "100 99 98 97 96 95 94 .... 4 3 2 1" which isn't sorted at all. It should be "1 2 3 4...98 99 100". The other inputs are just not sorted at all.

▲ 1 vote ● Reply ● Message ● Permalink

XandriethXs Feb 18, 2018 at 08:46 PM

Nope.... Check my solution. They are correct....

▲ 0 votes ● Reply ● Message ● Permalink

XandriethXs Feb 18, 2018 at 08:48 PM

|| Java Solution ||

```
import java.util.*;
class TestClass {
public static void main(String args[] ) throws Exception {
Scanner s = new Scanner(System.in);
int N = s.nextInt();
int A[] = new int[N];
int P[] = new int[N];
int B[] = new int[N];
int p = 0;
for (int i=0;i<N;i++)
{
A[i]=s.nextInt();
B[i]=A[i];
}
for (int i=0;i<N;i++)
{
int temp = A[i];
int j=i;
for(;(j>0 && temp<A[j-1]);j--)
{
A[j]=A[j-1];
}
A[j]=temp;
}
System.out.println();
for(int i=0;i<N;i++)
{
//int p;
for(int j=0;j<N;j++)
{
if(B[i]==A[j])
{
p = j+1;
break;
}
}
P[i]=p;
System.out.print(P[i] + " ");
}
}
}
```

▲ 1 vote ● Reply ● Message ● Permalink

Mohan Rathore Apr 22, 2017 at 02:28 PM

guys if you have prob then try this

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
int n,i,j,temp;
cin>>n;
int arr[n],brr[n];
for(i=0;i<n;i++)
{
cin>>arr[i];
brr[i]=arr[i];
}
for(i=0;i<n;i++)
{
int i, key, j;
for (i = 1; i < n; i++)
{
key = arr[i];
j = i-1;
while (j >= 0 && arr[j] > key)
{
arr[j+1] = arr[j];
j = j-1;
}
arr[j+1] = key;
}
}

for(i=0;i<n;i++){
for(j=0;j<n;j++){
if(brr[i]==arr[j])
cout<<j+1<<" ";
} } return 0;
}
```

▲ 1 vote ● Reply ● Message ● Permalink

Harish Raghav Sep 25, 2017 at 05:35 AM

all the 3 cases are getting failed when i submit, but when i copy the input of each case and paste them in "provide custom input", it is getting passed, where is the issue

```
#include <iostream>
using namespace std;
int main()
{
// read input :- size & array data
int nSize = 0;
cin>>nSize;

int nArray[nSize] = {0};
for(int nIdx = 0; nIdx < nSize; nIdx++)
{
cin>>nArray[nIdx];
}
// sort the data
int nTemp = 0, nTempIdx = 0;
for(int nIdx = 0; nIdx < nSize; nIdx++)
{
nTemp = nArray[nIdx];
nTempIdx = nIdx;

while((nTempIdx > 0) && (nTemp < nArray[nTempIdx - 1]))
{

nArray[nTempIdx] = nArray[nTempIdx-1];
nTempIdx -= 1;
}
nArray[nTempIdx] = nTemp;
}

// display data
for(int nIdx = 0; nIdx < nSize; nIdx++)
{
cout<<nArray[nIdx]<<" ";
}
//cout << "Hello World!" << endl;
return 0;
}
```

▲ 1 vote ● Reply ● Message ● Permalink

Harish Raghav Sep 27, 2017 at 06:53 AM

expecting a response this issue

▲ 0 votes ● Reply ● Message ● Permalink

Chamoda Pandithage Mar 01, 2018 at 06:02 AM

Can someone point out what's wrong with my code. It fails last 2 tests. (Python 3)

```
'''
count = input()
A = list(map(int, input().split(" ")))
I = list(range(1, len(A) + 1))
for i in range(0, len(A)):

if i > 0:

for j in list(reversed(range(0, i))):

if A[j] > A[j + 1]:

temp1 = A[j]

A[j] = A[j + 1]

A[j + 1] = temp1

temp2 = I[j]

I[j] = I[j + 1]

I[j + 1] = temp2

print(" ".join(map(str, I)))
```

▲ 1 vote ● Reply ● Message ● Permalink

Ketul Shah Jul 14, 2018 at 02:16 AM

My c++ solution:-

```
#include<bits/stdc++.h>
#define F(i,a,b) for(int i = (int) a ; i < (int) b ; ++i)
using namespace std;
int main(){
int N,temp;
cin >> N;
vector <int> arr;
vector <int> tarr;
F(i,0,N){
cin >> temp;
arr.push_back(temp);
}
tarr = arr;
F(i,0,N){
temp = arr[i];
int j = i;
while(j > 0 && arr[j-1] > temp){
arr[j] = arr[j-1];
j --;
}
arr[j] = temp;
// cout << i+j+1 << " ";
}
cout << endl;
F(i,0,N){
F(j,0,N){
if(tarr[i] == arr[j]){
cout << j+1 << " ";
break;
}
}
}


return 0;
}
```

▲ 1 vote ● Reply ● Message ● Permalink

Pranav Sarv Pathak Jan 31, 2017 at 01:52 PM

```
//an important topic of binary insertion sort
#include<stdio.h>
int bs(int a[],int x,int l,int u)
{
int m=l+((u-l)>>1);
if(l>=u) return (x>a[l])?l+1:l;
if(x==a[m]) return m;
if(x>a[m]) return bs(a,x,m+1,u);
return bs(a,x,l,m-1);
}
void insertionSort(int a[],int n)
{
int i,j,t,pos;
for(i=1;i<n;i++)
{
t=a[i]; j=i;
pos=bs(a,t,0,j);
while(j>pos)
{
a[j]=a[j-1];
j--;
}
a[j]=t;
}
}
int main()
{
int n,i,l,u,m;
scanf("%d",&n);
int a[n],p[n];
for(i=0;i<n;i++)
{
scanf("%d",a+i);
p[i]=a[i];
}
n=sizeof(a)/sizeof(a[0]);
insertionSort(a,n);
for(i=0;i<n;i++)
printf("%d ",bs(a,p[i],0,n-1)+1);
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Naveen Shukla Feb 16, 2017 at 07:55 AM

All inputs and outputs are corrects:

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

int N= s.nextInt();

int a[]= new int[N];
int indexa[]= new int[N];
for(int i=0;i<N;i++)
{
a[i]=s.nextInt();
indexa[i]=a[i];

}


for(int i=0;i<N;i++)
{
int temp=a[i];

int j=i;
while(j>0 && temp<a[j-1])
{
a[j]=a[j-1];

j=j-1;
}
a[j]=temp;

// j=index;
}
for(int i=0;i<indexa.length;i++)
{
for(int j=0;j<a.length;j++)
{
if(indexa[i]==a[j])
{
indexa[i]=j+1;
}
}
}

for(int array1:indexa)
{
System.out.print(array1+" ");
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Honoya Mar 02, 2017 at 05:19 AM

2º and 3º corrects outputs are not right. Only first one is right!
Tried in C++

▲ 0 votes ● Reply ● Message ● Permalink

Sreerag Nair Jun 11, 2017 at 06:44 AM

Look above for Ajish Athrayil's explanation. Try solving it that way and u will get the answer right.
What u basically have to do is this :
1. sort the array.
2. get the position of each element in the sorted array.
3. print the position of the element as per the question.

eg.
3
7 2 3
sorted output : 2 3 7
position : 1 2 3
output : 3 1 2

▲ 0 votes ● Reply ● Message ● Permalink

Narendar kumar Mar 31, 2017 at 02:47 PM

```
#include <iostream>
using namespace std;
int main()
{
int N;
cin>>N;
int arr[1000];
int brr[1000];
for(int i=0;i<N;i++)
{
cin>>arr[i];
}
for(int j=0;j<N;j++)
{
int temp=arr[j];
int k=j;
while(k>0 && temp<arr[k-1])
{
brr[k]=arr[k-1];
k=k-1;
}
brr[k]=temp;
}
for(int p=0;p<N;p++)
{
```

```
int q=arr[p];
for(int r=0;r<N;r++)
{
if(brr[r]==q)
{
cout<<(r+1)<<" ";
}
}
}
return 0;
}
```

What is wrong in this code,can anyone help me?

▲ 0 votes ● Reply ● Message ● Permalink

Atul Tiwari Jun 10, 2017 at 02:24 PM

Can it be done in O(N) or O(NlogN) ??

▲ 0 votes ● Reply ● Message ● Permalink

Susheendar Vekatachalam Jul 08, 2018 at 05:00 AM

Obviously no ,the sort itself in O(N^2)

▲ 0 votes ● Reply ● Message ● Permalink

Gopinath J Jul 06, 2017 at 01:35 PM

```
#include <stdio.h>
int main()
{
int n,i,j,temp,*a,*b;
scanf("%d",&n);
a=(int *)calloc(n,sizeof(int));
b=(int *)calloc(n,sizeof(int));
for(i=0;i<n;i++){
scanf("%d",&a[i]);
j=i;
b[i]=a[i];
temp=a[i];
while(j>0 && temp<b[j-1]){
b[j]=b[j-1];
j--;
}
b[j]=temp;
}
for(i=0;i<n;i++){
for(j=0;j<n;j++){
if(a[i]==b[j])
printf("%d ",j+1);
}
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Raghav Ravi Prakash Jul 11, 2017 at 02:48 PM

Java Solution

```
import java.util.*;
public class InsertionSort
{
public static Map<Integer,Integer> map = new HashMap<Integer,Integer>();

public static void main(String[] args)
{
Scanner in = new Scanner(System.in);
int size = in.nextInt();

int[] array = new int[size+1];
array[0] = 0;

for(int i = 1; i < array.length; i++)
array[i] = in.nextInt();

int[] copyArray = new int[array.length];
for(int i = 0; i < array.length; i++)
copyArray[i] = array[i];

sort(array);
mapKeysToIndices(array);
printIndices(copyArray);
}

public static void sort(int[] array)
{
for(int i = 2; i < array.length; i++)
{
int key = array[i];
int j = i - 1;

while(j >= 1 && array[j] > key)
{
array[j+1] = array[j];
j--;
}
array[j+1] = key;
}
}

private static void mapKeysToIndices(int[] sortedArray)
{
for(int i = 1; i < sortedArray.length; i++)
map.put(sortedArray[i],i);
}

public static void printIndices(int[] array)
{
for(int i = 1; i < array.length; i++)
System.out.print(map.get(array[i]) + " ");

System.out.println();
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Devesh kamboj Jul 16, 2017 at 06:14 AM

Its that easy:

```
#include <iostream>
using namespace std;
int main()
{
int n,i,j,temp,temp1;
cin>>n;
int a[n],b[n],c[n];

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

for(i=0;i<n;i++)
{
b[i]=a[i];
}


for(i=1;i<n;i++)
{
for(j=i-1;j>=0;j--)
{
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;

}
}
}

int k=0;

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

for(j=0;j<n;j++)
{
if(b[i]==a[j])
cout<<j+1<<" ";


}
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Vikas Gautam Jul 18, 2017 at 12:47 PM

this is working fine

```
#include <stdio.h>
int InsertionSort(int A[], int N)
{
int temp,i,j;


for(i=0;i<N;i++)
{
temp = A[i];
j = i;

while(j>0 && temp<A[j-1])
{
A[j]=A[j-1];
j = j-1;

}
A[j]=temp;
}


}
int main()
{
int *A,*A2,i,j,N;
scanf("%d",&N);
A = (int *)malloc(N*sizeof(int));
A2 = (int *)malloc(N*sizeof(int));
for(i=0;i<N;i++)
{
scanf("%d",&A[i]);
}
for(i=0;i<N;i++)
{
A2[i]=A[i];
}

InsertionSort(A,N);

for(i=0;i<N;i++)
{
for(j=0;j<N;j++)
{
if(A2[i]==A[j])
{
printf("%d ",j+1);
}
}
}


return 0;
}
```

0 votes ● Reply ● Message ● Permalink

Quangtri Thai Jul 19, 2017 at 11:42 AM

Here's what I came up with, sorting before searching, but is there a way to do this with only 1 array and only sorting?

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
int N;
cin >> N;

vector<int> A(N);
vector<int> AIndex(N);
for(int i = 0; i < N; i++)
{
cin >> A[i];
AIndex[i] = A[i];
}
for(int i = 0; i < N; i++)
{
int temp = A[i];
int j = i;
int tempIndex = 0;

while(j > 0 && temp < A[j - 1])
{
A[j] = A[j - 1];
j--;
}

A[j] = temp;
}

for(int i = 0; i < N; i++)
{
for(int k = 0; k < N; k++)
{
if(A[k] == AIndex[i])
{
AIndex[i] = k + 1;
break;
}
}
}

for(int i = 0; i < N; i++)
{
cout << AIndex[i] << " ";
}

return 0;
}
```

▲ 0 votes • Reply • Message • Permalink

Rahul Raj Jul 22, 2017 at 01:14 AM

Input:
100
100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

For the above input, correct output is mentioned as:
100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

which is wrong, but expected output is:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

My code gives above output. It is also correct as per the tutorial given above.
When I submit the solution, it says wrong output. whats wrong here?

▲ 0 votes • Reply • Message • Permalink

Mohit Kumar Aug 29, 2017 at 04:42 AM

You Have to give the location of the element in the sorted array while you need to take element from the original arrays.For Ex- Input Array has 100 as first element then you have to tell the location of 100 in the sorted array which is 100 and then 99 from the original array and so on

▲ 0 votes • Reply • Message • Permalink

Abhishek Ranjan Edited Jul 30, 2017 at 11:33 PM

passed all test cases .Explanation:->we just need to sort the elements and compare with the unsorted elements
and when we get the comparison just print the position+1 of the element.

```
#include<stdio.h>
int main()
{
int n;
int i,j;
int a[1000000];
int b[1000000];
scanf("%d",&n);
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);

}
for(int i=0;i<n;i++)
{
b[j]=a[i];
j++;
}
for(int i=1;i<=n-1;i++)
{
int value=a[i];
int hole=i;
while(hole>0 && a[hole-1]>value)
{
a[hole]=a[hole-1];
hole--;
}
a[hole]=value;
}
for(int j=0;j<n;j++)
{
for(int i=0;i<n;i++)
{
if(b[j]==a[i])
{
printf("%d ",i+1);
}
}
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Sanjeev Sharma Aug 14, 2017 at 12:43 PM

```
def func (arr) :
orig_arr = arr
for i in range(1,len(arr)) :
key = arr[i]
j = i-1
while (j>=0 and key<arr[j]) :
arr[j+1] = arr[j]
j = j-1
arr[j+1] = key
for key in orig_arr :
for i in range(len(arr)) :
if (arr[i] == key) :
```

```
print (i+1 ,end = " ")
break

n = int(input())
lis = list(map(int ,input().split()))
func (lis)
```

whats wrong here ?

▲ 0 votes ● Reply ● Message ● Permalink

Mohit Kumar Aug 29, 2017 at 04:37 AM

nice probelm!!

▲ 0 votes ● Reply ● Message ● Permalink

Himanshu Tomar Sep 01, 2017 at 02:21 PM

can anyone explain ....what is the use of inertion sort here i have simply used hashing and done it

▲ 0 votes ● Reply ● Message ● Permalink

Hemant Singh Patwal ✎ Edited Sep 06, 2017 at 12:36 AM

I think answers to test cases are incorrect

▲ 0 votes ● Reply ● Message ● Permalink

RICHA CHOUDHARY Sep 08, 2017 at 06:15 PM

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

sort(a, a+n);
for(int i=0; i<n; i++)
{
for(int j=0; j<n; j++)
{
if(b[i]==a[j])
{
cout<<j+1<<" ";
break;
}

}
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Sudamalla Moulica Sep 15, 2017 at 05:07 AM

h

▲ 0 votes ● Reply ● Message ● Permalink

Akshive Pandey Sep 26, 2017 at 03:41 AM

What's wrong with it

```
#include <stdio.h>
int binSearch(int *arr, int item, int size){
int lo = 0, hi = size-1;
while(hi > lo){
int mid = (lo+hi)/2;
if(*(arr+mid) == item){
return mid;
}
else if(*(arr+mid) > item){
hi = mid-1;
}
else{
lo = mid+1;
}
}
}
void insertionSort(int *arr, int size){
for(int i = 1; i < size; i++){
int j = i;
while(j > 0 && (*(arr+j) < *(arr+j-1))){
int temp = *(arr+j-1);
*(arr+j-1) = *(arr+j);
*(arr+j) = temp;
j = j-1;
}
}
}
int main()
{
int N;
scanf("%d", &N);
int *arr = (int *)malloc(sizeof(int)*N);
int *arr2 = (int *)malloc(sizeof(int)*N);
for(int i = 0; i < N; i++){
scanf("%d", (arr+i));
*(arr2+i) = *(arr+i);
}

insertionSort(arr, N);

for(int i = 0; i < N; i++){
int item = *(arr2+i);
int res = binSearch(arr, item, N);
printf("%d ", res+1);
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Aman Bedi Oct 09, 2017 at 01:21 PM

```
N=int(raw_input())
arr = raw_input().split()
arr = [int(a) for a in arr]
t=arr[:]
for i in range(N):
temp = arr[i]
j=i
while(j>0 and temp<arr[j-1]):
arr[j]=arr[j-1]
j=j-1
arr[j]=temp
```

```
for i in range(len(t)):
print("%d" % (arr.index(t[i])+1)),
```

▲ 0 votes ● Reply ● Message ● Permalink

Konstantin Chukharev Nov 10, 2017 at 07:07 AM

```
def sorted1(a):
b=list(a)
for i in range(1,N):
temp=b[i]
j = i
while(j>0 and temp<b[j-1]):
b[j]=b[j-1]
j-=1
b[j]=temp
return b
N = int(input())
a = list(map(int, input().split()))
print(*[sorted1(a).index(i)+1 for i in a])
```

PS the def part is useless, as built-in sorted function is better:)

▲ 0 votes ● Reply ● Message ● Permalink

Jeet Holmes Oct 09, 2017 at 05:02 PM

Managed to solve all test cases and also avoided extra for loop to create map for keys and values by making map on the fly while sorting. This is JAVA code:

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String line = br.readLine();
String[] firstLine = line.split(" ");

int N = Integer.parseInt(firstLine[0]);

line = br.readLine();
String[] secondLine = line.split(" ");
int numbers[] = new int[N];
for(int i=0;i<secondLine.length;i++) {
numbers[i] = Integer.parseInt(secondLine[i]);
}
int[] unsorted = new int[N];
System.arraycopy(numbers, 0, unsorted, 0, N);

//Insertionsort IMPL

HashMap<Integer, Integer> result = new HashMap<>();
int j;

for(int i=0;i<N;i++) {
int temp = numbers[i];
j=i;

result.put(numbers[i], i+1);

while(j>0 && temp<numbers[j-1]) {
numbers[j] = numbers[j] + numbers[j-1];
numbers[j-1] = numbers[j] - numbers[j-1];
numbers[j] = numbers[j] - numbers[j-1];
j = j-1;
result.put(numbers[j], j+1);
result.put(numbers[j+1], j+2);
}
}
```

```
for(int unsortedPos: unsorted) {
System.out.print(result.get(unsortedPos)+" ");
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Bharat Kumar Oct 28, 2017 at 10:33 AM

```
// what is problem in thise code plz give me answer
#include <iostream>
using namespace std;
int main()
{int n;
cin>>n;
int a[n];
for(int i=0;i<n;i++){
cin>>a[i];
}
for(int i=0;i<n;i++)
{
if(i==0)
{
int j;
if(a[i]>a[i+1])
{ j=i;
while(j!=(n-1))
{
if(a[j]>a[j+1])
{ int t;
t=a[j];
a[j]=a[j+1];
a[j+1]=t;
j++;
}
else
{
a[i]=a[i];
j++;
}
}
}
}
else
{
int j;
if(a[i]>a[i+1])
{ j=i;
while(j!=(n-1))
{
if(a[j]>a[j+1])
{ int t;
t=a[j];
a[j]=a[j+1];
a[j+1]=t;
j++;
}
else
{
a[i]=a[i];
j++;
}
}
}

if(a[i]<a[i-1])
{
while(i!=0)
{
if(a[i]<a[i-1])
{
int t;
t=a[i];
a[i]=a[i-1];
a[i-1]=t;
}
else
{
a[i]=a[i];
}
i--;
}
}
}
}
for(int i=0;i<n;i++){
cout<<a[i]<<" ";
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

md anwaruzzaman Oct 31, 2017 at 03:40 PM

What is wrong with this piece of code

```
#include <iostream>
using namespace std;
int main()
{
//cout << "Hello World!" << endl;
int n,a[100],b[100];
cin>>n;
for(int i=0;i<n;i++)
{
cin >> a[i];
}
for(int k=0;k<n;k++)
{ b[k]=a[k];
int temp=a[k];
int j=k;
while(j>0 && temp<a[j-1])
{
a[j]=a[j-1];
j=j-1;
//b[k]=j+1;

}
a[j]=temp;
}

for(int m=0;m<n;m++)
{
```

```
for(int i=0;i<n;i++)
{
if(a[m]==b[i])
cout<< (i+1) <<" ";
}

}
return 0;


}
```

▲ 0 votes ● Reply ● Message ● Permalink

md anwaruzzaman Edited Nov 02, 2017 at 11:23 AM

@admin - The solution gives correct result with custom input but it gives wrong answer with submission .. please note here the custom input is same as input #1,#2,#3 of the submission. Please reply

code is given below

```
#include <iostream>
using namespace std;
int main()
{
//cout << "Hello World!" << endl;
int n,a[100],b[100];
cin>>n;
for(int i=0;i<n;i++)
{
cin >> a[i];
}
for(int f=0;f<n;f++)
{
b[f]=a[f];
}
for(int k=0;k<n;k++)
{
int temp=a[k];
int j=k;
while(j>0 && temp<a[j-1])
{
a[j]=a[j-1];
j=j-1;
//b[k]=j+1;

}
a[j]=temp;
}

for(int m=0;m<n;m++)
{
for(int i=0;i<n;i++)
{
if(a[m]==b[i])
cout<< (i+1) <<" ";
}

}
return 0;


}
```

▲ 0 votes ● Reply ● Message ● Permalink

SATYAJEET BEHERA Dec 10, 2017 at 06:24 AM

Make array size to 101 that might work !

▲ 0 votes ● Reply ● Message ● Permalink

Semih Chelik Dec 13, 2017 at 09:12 PM

```
!!!WORKED!!!
#include <iostream>
#include <cmath>
#include <stdio.h>
using namespace std;
#include <iostream>
#include <cmath>
#include <stdio.h>
using namespace std;
void insertion_sort ( int A[ ] , int n)
{
for( int i = 0 ;i < n ; i++ ) {
/*storing current element whose left side is checked for its
correct position .*/
int temp = A[ i ];
int j = i;
/* check whether the adjacent element in left side is greater or
less than the current element. */
while( j > 0 && temp < A[ j -1]) {
// moving the left side element to one position forward.
A[ j ] = A[ j-1];
j= j - 1;
}
// moving current element to its correct position.
A[ j ] = temp;
}
}

int main()
{
int size;
cin>>size;
int nums[size-1];
for(int i = 0;i<size;i++)
{
cin>>nums[i];
}
insertion_sort(nums,size);
for(int j=0;j<size;j++)
{
cout<<nums[j]<<" ";
}

return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Ritvik Kona ✎ Edited Dec 15, 2017 at 07:09 AM

The insertion sort algorithm is working, I am unable to execute the second part of the problem of matching the indexes of the original array to that of the sorted array.Could anyone suggest an edit
PYTHON PROGRAM --->

```
def insertion_sort(A,n):
for i in range(n):
temp = A[i]
j = i
while j > 0 and temp < A[j-1]:
A[j] = A[j-1]
j -= 1
A[j] = temp
return A
n = int(input())
lst = input().split()
lst1 = []
for item1 in lst:
lst1.append(item1)
rlst = insertion_sort(lst,n)
rlst1 = [rlst.index(i)+1 for i in lst1 ]
result = ""
for item in rlst:
result += str(item) + " "
print(result)
```

▲ 0 votes ● Reply ● Message ● Permalink

Prakash Yadav Dec 20, 2017 at 06:28 AM

```
#include <stdio.h>

int main()
{
int n,i,j,temp;
scanf("%d",&n);
int a[n],b[n];
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
b[i]=a[i];
}
for(i=0;i<n;i++)
{
temp=a[i];
j=i;
while(j>0 && temp<a[j-1])
{
a[j]=a[j-1];
j=j-1;
}
a[j]=temp;
}
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
if(b[i]==a[j]) printf("%d ",j+1);
}
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Puneet Singh Dec 20, 2017 at 08:56 AM

```
public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
int[] arr = new int[scan.nextInt()];
for (int i = 0; i < arr.length; i++) {
arr[i] = scan.nextInt();
}
int[] indexArr = insertionSort(arr);
for (int i : indexArr) {
System.out.print(i + 1 + " ");
}
scan.close();
}
private static int[] insertionSort(int[] arr) {
int[] index = new int[arr.length];
int[] aux = new int[arr.length];

for (int i = 0; i < index.length; i++) {
index[i] = i;
aux[i] = i;
}
for (int i = 1; i < arr.length; i++) {
for (int j = i - 1; j >= 0; j--) {
if (arr[j + 1] < arr[j]) {
swap(arr, j, j + 1);
swap(index, j, j+1);
swap(aux, index[j], index[j+1]);
}
}
}
return aux;
}
private static void swap(int[] arr, int i, int j) {
int temp = arr[i];
arr[i] = arr[j];
arr[j] = temp;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Bushera Mustofa Dec 21, 2017 at 02:41 AM

hey here am submitted the code just like below but take care of even for space okey..........i failed so much time for only space

```
#include <iostream>
using namespace std;
int k=0;
void insesort(int A[],int n)
{
for(int i=1;i<n;i++)
{
int temp=A[i];
int j=i-1;
while(j>=0 && A[j]>temp)
{
A[j+1]=A[j];
j--;
}
A[j+1]=temp;
}
}
void postion(int array[],int ID[],int m)
{
for(int l=0;l<m;l++)
{
for(int j=0;j<m;j++)
{
if(array[l]==ID[j])
cout<<j+1<<" ";
}
}
}
int main()
{
int n;

cin>>n;
int id[n],tempo[n];
for(int i=0;i<n;i++)
{
cin>>id[i];
tempo[i]=id[i];
}
insesort(id,n);
postion(tempo,id,n);
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

abhinav prabandham Dec 28, 2017 at 04:33 AM

all test cases are correct

▲ 0 votes ● Reply ● Message ● Permalink

Rishma Jan 14, 2018 at 02:20 AM

```
#include<iostream>
using namespace std;
int main()
{
int N,i,j,t,s[200];
cin>>N;
int a[N];

for(i=0;i<N;i++)
{
cin>>a[i];
}
//copy a to s
for(i=0;i<N;i++)
{
s[i]=a[i];
}
for(i=0;i<N;i++)
{
j=i;
t=a[i];
while(j>0&&t<a[j-1])
{
a[j]=a[j-1];
j=j-1;
}
a[j]=t;
}
/*for(i=0;i<N;i++)
{
cout<<a[i]<<" ";
}*/
for(i=0;i<N;i++)
{
for(j=0;j<N;j++)
{
if(s[i]==a[j])
{
cout<<j+1<<" ";
}
}
}

}
```

▲ 0 votes ● Reply ● Message ● Permalink

Pradhan Rishi Sharma Feb 07, 2018 at 01:18 AM

Correct Solution

```
import java.util.Scanner;
class TestClass{
public static void main(String[] args) throws Exception{
int size;
Scanner sc = new Scanner(System.in);
size = sc.nextInt();
int arr[] = new int[size];
int auxarr[] = new int [size];
for (int i=0;i<size;i++){
arr[i] = sc.nextInt();
auxarr[i] = arr[i];
}
for (int j = 0; j < size; j++) {
int temp = arr[j];
int k = j;
/*storing current element whose left side is checked for correct position*/
/*Check whether the adjacent element in left is greater or less than current element*/
while ( k > 0 && temp < arr[k-1]){
// moving left side element to one space forward
arr[k] = arr[k-1];
k -= 1;
}
// moving current element to correct position
arr[k] = temp;
}
for (int l = 0; l < size; l++) {
int showele = auxarr[l];
for (int m = 0; m < size; m++) {
if (arr[m] == showele){
System.out.print(m+1+" ");
}
}
}
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Sirisha K Mar 06, 2018 at 12:31 AM

```
N=int(input())
a=input()
a=[int(x) for x in a.split(' ')]
for i in range(N):
for j in range(i-1):
if a[j]<a[i]:
a[i],a[j]=a[j],a[i]
a=sorted(a)
for e in a:
print(e,end=" ")
```

▲ 0 votes ● Reply ● Message ● Permalink

krishnakanth Reddy kosnam Jun 20, 2018 at 07:28 AM

```
#include <iostream>
using namespace std;
void swap(int &a, int &b){
a = a^b;
b = a^b;
a = a^b;
}
int main(){

int size;
cin >>size; // Reading input from STDIN
int array[size];
int position[size];
for(int i = 0;i<size;i++){
cin>>array[i];
position[i] = array[i];

}
for (int i = 0 ;i<size;i++){
for (int j = i;j>0;j--){
while(array[j]<array[j-1]){
swap(array[j],array[j-1]);
}
}
}


for(int i = 0;i<size;i++){
for(int j = 0;i<size;j++){
if(position[i] == array[j]){
cout << j+1 <<" ";
break;
}
}
}

}
```

▲ 0 votes ● Reply ● Message ● Permalink

Vaibhav Physics Jul 04, 2018 at 08:54 AM

this one also works guys

```
number = int(input())
a = list(map(int,input().strip().split()))
for i in range(number):
for j in range(number):
if a[i]<a[j]:
```

```
a[i],a[j] = a[j],a[i]
print(*a)
```

▲ 0 votes • Reply • Message • Permalink

Manojeet Das Jul 12, 2018 at 01:15 AM

Easiest Solution:

```
using namespace std;
#include<bits/stdc++.h>
int main()
{
int n;
cin>>n;
int a[n],b[n];
for(int i=0;i<n;i++)
{
cin>>a[i];
b[i]=a[i];
}
map<int,int> m;
sort(b,b+n);
for(int i=0;i<n;i++)
{
m[b[i]]=i+1;
}
for(int i=0;i<n;i++)
{
cout<<m[a[i]]<<" ";
}


}
```

▲ 0 votes • Reply • Message • Permalink

Дима Гурин Jul 20, 2018 at 04:21 AM

```
#include <iostream>
#include <vector>
#include <algorithm>
//
using VecI = std::vector<int>;
using std::cin;
using std::cout;
using std::endl;
void insertionSort(VecI &vec, int sz) noexcept {
int tmp;
for (int i = 0; i < sz; i++) {
tmp = vec[i];
int j = i;
while(j > 0 && tmp < vec[j - 1]) {
vec[j] = vec[j - 1];
j--;
}
vec[j] = tmp;
}
}
void output(VecI &unsorted, VecI &sorted, int sz) noexcept {
if (std::equal(unsorted.cbegin(), unsorted.cend(), sorted.cbegin()) || std::equal(unsorted.cbegin(),
unsorted.cend(), sorted.crbegin())) {
for (auto n: unsorted)
cout << n << " ";
}
else {
for (int i = 0; i < sz; i++) {
cout << std::distance(sorted.cbegin(), std::find(sorted.cbegin(), sorted.cend(), unsorted[i])) + 1<< " ";
}
}
}
int main(void) {
int sz; cin >> sz;
VecI vec(sz, 0);
for (int i = 0; i < sz; i++) {
int tmp; cin >> tmp;
vec[i] = tmp;
}
VecI sorted_vec = vec;
insertionSort(sorted_vec, sz);
output(vec, sorted_vec, sz);
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Wate Soyan ✎ Edited Jul 23, 2018 at 03:23 AM

my python3 solution:

```
def argsorted(a):
n = len(a)
ix_ = list(range(n))
for i in range(1, n):
tmp = a[i]
tmp_ix = ix_[i]
j = i
pre_j = j - 1
while j > 0 and a[ix_[pre_j]] > tmp:
ix_[j] = ix_[pre_j]
j = pre_j
pre_j -= 1
ix_[j] = tmp_ix
return ix_

ix_ = argsorted(A)
map_ = {ix_[k]:k for k in range(N)}
rank = [map_[k]+1 for k in range(N)]
res = str(rank[0])
for c in rank[1:]:
res += " "
res += str(c)
```

▲ 0 votes ● Reply ● Message ● Permalink

Rohan Shenoy Aug 04, 2018 at 09:08 AM

The sample test cases can be better explained

▲ 0 votes ● Reply ● Message ● Permalink

About Us | Innovation Management | Technical Recruitment
University Program | Developers Wiki | Blog
Press | Careers | Reach Us

Site Language: English ▼ | Terms and Conditions | Privacy |© 2018 Hacke ch