Insertion Sort Compare Metric

Problem Code: **COMPSWAP**

All Submissions
(/DI15R066/status/COMPSWAP)

Successful Submissions +

Problem description

Insertion sort is a simple sorting algorithm that builds the final sorted array one item at a time.
Given below is pseudocode of insertion sort algorithm applied on array A (zero-based indexing).

```
       def compare(a, b) :
           if a > b return 1
           else return -1

       for i : 1 to length(A)
           j = i - 1
           while j > 0
              if compare(A[j-1], A[j]) > 0
                   swap(A[j], A[j-1])
                   j = j - 1
              else break
```

Given an array **A** of distinct integers , find difference between number of compare function calls and number of swap function calls by above algorithm when applied on **A**.

Let us take an example with A = **{1, 2, 4, 3}**. If we apply insertion sort as above on A we call sequeunce of compare and swap functions in following order

```
   compare (A[0], A[1])
   compare (A[1], A[2])
   compare (A[2], A[3])
   swap       (A[2], A[3])
   compare (A[1], A[2])
```

Here compare function is called 4 times, swap function is called 1 time. The answer is 4-1 = 3.

Input

- The first line of the input contains an integer **T** denoting the number of test cases.T test cases follow
- The first line of each test case contains a single integer **N** denoting length of array **A**.
- The second line of each test case contains **N** space-separated distinct integers $\mathbf{A_0}$, $\mathbf{A_1}$, ..., $\mathbf{A_{N-1}}$ denoting the elements of **A**.

Output

- For each test case, output a single line containing difference between number of compare function calls and number of swap function calls.

Constraints

- **1 ≤ T ≤ 100**
- **1 ≤ N ≤ 10000**
- **1 ≤ A[i] ≤ N**

Example

```
Input:
6
2
1 2
2
2 1
4
1 2 4 3
4
2 3 1 4
4
4 3 2 1
10
5 3 2 4 1 6 7 9 8 10

Output:
1
0
3
2
0
6
```

Explanation

Example test case 1.

For **A = {1, 2}** following is the sequence of compare and swap functions called.

```
   compare (A[0], A[1])
```

Hence answer is 1 for first test case.

Example test case 2.

For **A = {2, 1}** following is the sequence of compare and swap functions called.

```
   compare (A[0], A[1])
   swap       (A[0], A[1])
```

Hence answer is 0 for second test case.
Example test case 4.

For **A = {2, 3, 1, 4}** following is the sequence of compare and swap functions called.

```
   compare (A[0], A[1])
   compare (A[1], A[2])
   swap       (A[1], A[2])
   compare (A[0], A[1])
   swap       (A[0], A[1])
   compare (A[2], A[3])
```

Hence answer is 2 for fourth test case.

Note

The answer can be found using the given insertion sort pseudo code but will have O(N^2) time complexity.
Therefore we made sure O(N^2) solutions will not pass under the time limits.

Author:	directi_campus (/users/directi_campus)
Tags:	directi_campus (/tags/problems/directi_campus)
Date Added:	25-01-2015
Time Limit:	2 secs
Source Limit:	50000 Bytes
Languages:	C, CPP 4.3.2, CPP 6.3, CPP14, JAVA, PYTH, PYTH 3.5

Comments

The time now is: 10:22:48 PM
Your IP: 123.201.210.10

CodeChef (http://www.codechef.com) - A Platform for Aspiring Programmers

CodeChef was created as a platform to help programmers make it big in the world of algorithms, **computer programming** and **programming contests**. At CodeChef we work hard to revive the geek in you by hosting a **programming contest** at the start of the month and another smaller programming challenge in the middle of the month. We also aim to have training sessions and discussions related to **algorithms**, **binary search**, technicalities like **array size** and the likes. Apart from providing a platform for **programming competitions**, CodeChef also has various algorithm tutorials and forum discussions to help those who are new to the world of **computer programming**.

Practice Section (https://www.codechef.com/problems/easy) - A Place to hone your 'Computer Programming Skills'

Try your hand at one of our many practice problems and submit your solution in a language of your choice. Our **programming contest** judge accepts solutions in over 35+ programming languages. Preparing for coding contests were never this much fun! Receive points, and move up through the CodeChef ranks. Use our practice section to better prepare yourself for the multiple **programming challenges** that take place through-out the month on CodeChef.

Compete (https://www.codechef.com/problems/easy) - Monthly Programming Contests and Cook-offs

Here is where you can show off your **computer programming skills**. Take part in our 10 day long monthly coding contest and the shorter format Cook-off **coding contest**. Put yourself up for recognition and win great prizes. Our **programming contests** have prizes worth up to INR 20,000 (for Indian Community), $700 (for Global Community) and lots more CodeChef goodies up for grabs.

Programming Tools

Online IDE (https://www.codechef.com/ide)
Upcoming Coding Contests (http://www.codechef.com/contests#FurtureContests)
Contest Hosting (http://www.codechef.com/hostyourcontest)
Problem Setting (http://www.codechef.com/problemsetting)
CodeChef Tutorials (http://www.codechef.com/wiki/tutorials)
CodeChef Wiki (https://www.codechef.com/wiki)

Practice Problems

Easy (https://www.codechef.com/problems/easy)
Medium (https://www.codechef.com/problems/medium)
Hard (https://www.codechef.com/problems/Hard)
Challenge (https://www.codechef.com/problems/challenge)
Peer (https://www.codechef.com/problems/extcontest)
School (https://www.codechef.com/problems/school)
FAQ's (https://www.codechef.com/wiki/faq)

Initiatives

Go for Gold (http://www.codechef.com/goforgold)
CodeChef for Schools (http://www.codechef.com/school)
Campus Chapters (http://www.codechef.com/campus_chapter/about)
Domain Registration in India (http://www.bigrock.in/) and Web Hosting (http://www.bigrock.com/web-hosting/) powered by BigRock