Petr Mitrichev (competitive programmer) Codeforces +5

What is the solution to "Petr and permutation" (987E) on Codeforces?

http://codeforces.com/contest/987/problem/E

Answer Follow · 1 Request

1 Answer

Saikat Ghosh, Loves Mathematics

Updated Jun 3

> You are given a random permutation of the first n positive integers.
>
> The permutation started out as an identity permutation $[i = p(i)]$ and was then subjected to either $3n$ random swaps or $7n + 1$ random swaps.
>
> Determine whether the random permutation given has been swapped $3n$ times or $7n + 1$ times. (It is guaranteed one of these is the answer).

Question Link

At first sight, this question looks impossible ! How could we *ever* find out whether a permutation has been randomly shuffled $3n$ or $7n + 1$ times !

Fortunately a simple *mathematical observation* makes the problem a lot simpler!

The numbers $3n$ and $7n + 1$ always have different parity !

Proof -

s even, then n is even. This forces $7n + 1$ to be

If $3n$ is odd, then n is odd. This makes $7n + 1$ even.

This means we have to simply count the *parity* of the permutation - And match it with the parity of the number of operations !

The parity of a permutation is the number of inversions in it, i.e. the number of pairs x, y such that $x < y$ and $p(x) > p(y)$

Now, the question is, *how do we count the number of inversions of a permutation ?*

We can do it in brute force in $O(n^2)$ or with a segment tree in $O(n \log n)$ or we could just do it with cycle decomposition in $O(n)$!

[$O(n^2)$ will not be accepted but $O(n \log n)$ is fine.]

To do it with **cycle decomposition**, let us first construct a graph with n vertices and draw an edge in between every i and $p(i)$

- Swapping elements in different cycles creates a longer cycle.

Related Questions

Android (operating system): How many combinations does Android 9 point unlock have?

What it is like to meet or know Petr Mitrichev?

Would I have an opportunity to work for big tech companies like Microsoft, Google, Amazon etc. if I have knowledge of only competitive program...

How do I become a red coder on Codeforces and TopCoder in one year?

What does the constructive algorithms tag mean at Codeforces?

What is the practical real-life use of permutation and combination?

Should I do HackerRank, SPOJ, TopCoder, CodeForces or CodeChef?

What are some must-do problems on Codeforces?

What is the best strategy to improve my skills in competitive programming in C++ in 2-3 months?

What is the story behind your username at CodeForces?

+ Ask New Question

More Related Questions

Question Stats

1 Public Follower

1,120 Views

Edits

and b and the cycle size reduces by 2, then it would mean that a and b form a cycle amongst themselves ! This can only happen when there are 2 elements left.)

- The number of swaps required to put a cycle of size M back in place is $M - 1$

Here is the code for finding the number of swaps required to put a permutation back in place (*number of inversions*) through *cycle decomposition*.

```
int no_of_swaps = 0;
vector <int> visited(n + 1, false);
for(int i = 1; i <= n; i++)
{
    if(!visited[i])
    {
        int cycle_size = 0;
        for(int current = i; !visited[current]; current = permutation[cu
        {
            visited[current] = true;
            cycle_size++;
        }

        no_of_swaps += cycle_size - 1;
    }
}
```

To summarise -

- Find the parity of the permutation.
- Match parity of permutations with parity of swaps.

Here is my code.

761 Views · View Upvoters