



Algorithm Design

molecule

Kwangmin Kim

573 - 8001



molecule

Outline

- Problem
- Solution
- Code
- Result

Problem (Timus Online Judge)

1024. Permutations

We remind that the permutation of some final set is a one-to-one mapping of the set onto itself. Less formally, that is a way to reorder elements of the set.

For example, one can define a permutation of the set $\{1,2,3,4,5\}$ as follows:

$$P(n) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 1 & 5 & 2 & 3 \end{pmatrix}$$

What is the value of the expression $P(P(1))$? It's clear, that $P(P(1)) = P(4) = 2$. And $P(P(3)) = P(5) = 3$. One can easily see that if $P(n)$ is a permutation then $P(P(n))$ is a permutation as well. In our example (check it by yourself)

$$(Pn)^2 = P(P(n)) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 3 & 1 & 5 \end{pmatrix}$$

$$(Pn)^k = Pn.$$

$$k = ?$$

time limit = 2 sec



Solution

1. Using Recursion

1.1 Make input data form

2.2 Copy original permutation to other variable

2.3 Changing permutation by Recursive function

2.4 Compare between original permutation and changed permutation

2.5 Recursively, Repeat function

Therefore, it is too slow that over time limit



Solution

2. Using Number Cycle

- 1.1 Make input data form
- 2.2 Set variable of condition
- 2.3 Find all of number cycle in permutation
- 2.4 Calculate greatest common divisor
- 2.5 Calculate least common multiple
- 2.6 Get answer



Solution 1 Code

```
import copy,sys
sys.setrecursionlimit(640000)
n=input();
n=int(n);
per=raw_input();
per=map(int,per.split());
goal=copy.deepcopy(per);
def is_goal(n):
    global goal
    for i in range(len(goal)):
        if n[i]!=goal[i]:
            return False
    return True
def function(n,c):
    global form
    if c!=0 and is_goal(n):
        return c
    else:
        for i in range(len(n)):
            n[i]=goal[n[i]-1]
        c+=1
        return function(n,c)
print function(per,0)
```




Solution 2 Code

```
def gcd(a,b):
    while a%b!=0:
        a%=b;
        a,b=b,a
    return b;
n=input();
n=int(n);
per=raw_input();
per=map(int,per.split());
per=map(lambda x: x-1, per)
visited=[-1]*(n+1);
ans=1;
for i in range(n):
    if (visited[i]==0):
        continue;
    else:
        pos=i;
        sz=0;
        while(visited[pos]==-1):
            visited[pos]=0
            pos =per[pos]
            sz+=1
        max=gcd(ans, sz);
        ans=(ans/max)*(sz/max)*max;
print ans
```



molecule

Solution 1 Result

7382262	20:20:31 9 May 2017	kwangmin	1024. Permutations	Visual C++ 2013	Time limit exceeded
7382260	20:19:44 9 May 2017	kwangmin	1024. Permutations	Visual C++ 2013	Compilation error
7382259	20:19:33 9 May 2017	kwangmin	1024. Permutations	Visual C 2013	Compilation error
7382258	20:18:53 9 May 2017	kwangmin	1024. Permutations	Visual C 2013	Compilation error
7382257	20:18:17 9 May 2017	kwangmin	1024. Permutations	Visual C 2013	Compilation error
7382256	20:17:04 9 May 2017	kwangmin	1024. Permutations	Visual C++ 2013	Compilation error
7382052	17:36:16 9 May 2017	kwangmin	1024. Permutations	Python 2.7	Wrong answer
7380111	00:53:19 8 May 2017	kwangmin	1024. Permutations	Python 2.7	Time limit exceeded
7380108	00:47:10 8 May 2017	kwangmin	1024. Permutations	Python 2.7	Time limit exceeded



molecule

Solution 2 Result

7383024	11:59:44 10 May 2017	kwangmin	1024. Permutations	Visual C++ 2013	Accepted
7383023	11:58:19 10 May 2017	kwangmin	1024. Permutations	Python 2.7	Accepted
7382577	23:55:54 9 May 2017	kwangmin	1024. Permutations	Visual C++ 2013	Accepted
7382575	23:55:26 9 May 2017	kwangmin	1024. Permutations	Visual C++ 2013	Accepted
	23:53:11				