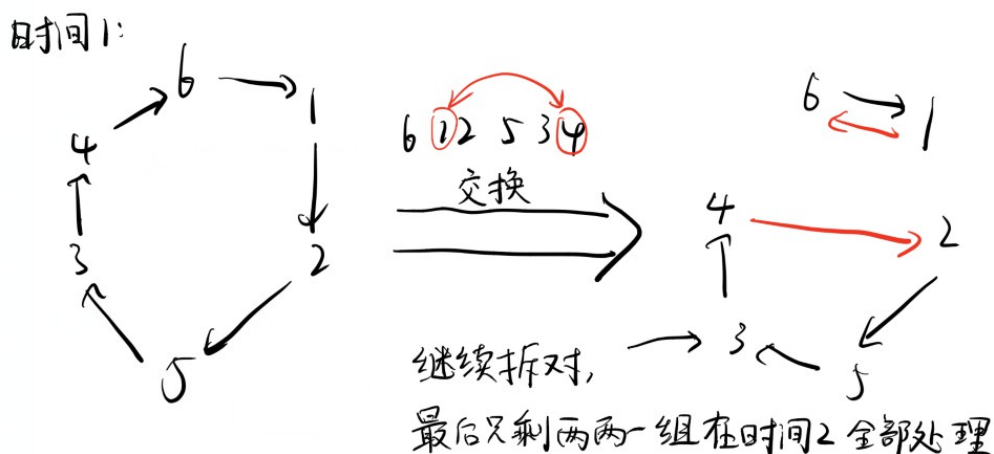


我们可以把这个置换划分为多个循环节，然后答案只和循环节有关。

什么是循环节？就是从某个下标  $i$  开始一直沿着  $a[i]$  走，直到走回  $i$ 。如 1, 4, 5, 3, 2（下标从 1 开始），从下标 2 开始走就是 4（下标 2）→ 3（下标 4）→ 5（下标 3）→ 2（下标 5）→ 4（下标 2）。这就是一个循环节，长度为 4。

回到题目中。我们首先忽略循环节长度为 0 的情况。然后讨论：

1. 循环节长度为 2。那么显然相互换一下就行了，次数时间都是 1。
2. 循环节长度大于 2。那么有：
  1. 如果要考虑让交换次数最少，（理论上）可以证明：设循环节长度为  $l$ ，那么次数至少为  $l - 1$ 。直观的理解是每一次交换至多让一个元素复位，而最后只剩两个元素时交换一次，两者就都可以复位。这样我们分别考虑每一个循环节，计数即可。
  2. 如果要是时间最少，我们可以构造出一种方案使得时间为 2。以 6, 1, 2, 5, 3, 4 演示：



循环节长度为奇数也是类似的处理方法。当然，本题不要求构造方案，所以只需要判断就行。

```
1 #include <bits/stdc++.h>
2 #define INF 2000000000
3 using namespace std;
4 typedef long long ll;
5 int read(){
6     int f = 1, x = 0;
7     char c = getchar();
8     while(c < '0' || c > '9'){if(c == '-') f = -f; c = getchar();}
9     while(c >= '0' && c <= '9')x = x * 10 + c - '0', c = getchar();
10    return f * x;
11 }
12 int n, a[1000005];
13 bool vis[1000005];
14 void init(){
15     for (int i = 1; i <= n; ++i)
16         a[i] = read();
17     memset(vis, 0, sizeof(vis));
```

```
18 }
19 void solve(){
20     int tm = 0, maxi = 0;
21     for (int i = 1; i <= n; ++i){
22         if (vis[i] || i == a[i]) continue;
23         int t = a[i], cnt = 1;
24         vis[i] = true;
25         while (t != i)
26             vis[t] = true, t = a[t], ++cnt;
27         maxi = max(maxi, (cnt == 2 ? 1: 2));
28         tm += (cnt == 2 ? 1: cnt - 1);
29     }
30     printf("%d\n%d\n", tm, maxi);
31 }
32 int main(){
33     while (scanf("%d", &n) == 1){
34         init();
35         solve();
36     }
37     return 0;
38 }
```