



# Minimum Swaps 2

Problem

Submissions

Leaderboard

Discussions

Editorial

Editorial by [rishi\\_07](#)

1. The idea is that if  $a$  occupies  $b's$  position,  $b$  occupies  $c's$  position and so on, then there will be some integer  $x$  which will occupy  $a's$  position. So, this forms a cycle.
2. So, if any element  $arr_i$  is not at its correct position, we shift it to its correct position  $j$ , then shift  $arr_j$  to its correct position  $k$  and so on. So, if  $len$  is the length of the cycle (number of elements in the cycle), then it will require a minimum of  $len - 1$  swaps to rearrange the elements of the cycle to their correct positions.
3. We find all such cycles and compute our answer.

The correct positions of all the elements can be found by sorting the array by value and keeping track of the old and new positions. You may gain more clarity by the setters solution.

Set by [rishi\\_07](#)

Problem Setter's code:

```
#include<bits/stdc++.h>
using namespace std;

int a[100005];
bool visited[100005];

int solve(int n)
{
    pair<int, int> p[n];

    for (int i = 0; i < n; i++)
    {
        p[i].first = a[i];

        // Storing the original position of a[i]
        p[i].second = i;
    }
}
```

## STATISTICS

Difficulty: **Medium**Time Complexity: **nlogn**Required Knowledge: **Array Manipulation, sorting**Publish Date: **Jun 22 2018**

This is a Practice Challenge

## NEED HELP?

[View discussions](#)[View top submissions](#)

```

    sort(p, p+n);
    int ans = 0;

    for (int i = 0; i < n; i++)
    {
        //visited[i]=true indicates that index i belongs
        to a cycle that is already counted

        //p[i].second = i denotes that the ith element w
        as at its correct position

        if (visited[i] || p[i].second == i)
            continue;

        int cycle_size = 0;
        int j = i;

        //Counting the size of the cycle
        while (!visited[j])
        {
            visited[j] = 1;
            j = p[j].second;
            cycle_size++;
        }

        ans += (cycle_size - 1);
    }

    return ans;
}

int main()
{
    int n;
    scanf("%d", &n);

    for(int i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }

    int ans = solve(n);
    printf("%d\n", ans);
    return 0;
}

```



Tested by [rock19](#)

Problem Tester's code:

```

vector<int>v[100003];
bool visit[100003];

// This function return the size of the cycle as mentioned in the explanation.
int dfs(int i)
{
    visit[i] = true;
    int z = 1;

    for(auto x: v[i])
        if(!visit[x])
            z += dfs(x);

    return z;
}

int minimumSwaps(vector<int> A) {

    for(int i = 0; i < A.size(); ++i )
        v[i].push_back(A[i]-1), v[A[i]-1].push_back(i);

    int c = 0;

    for(int i = 0; i < A.size(); ++i)
    {
        if(!visit[i])
            c += dfs(i) - 1;
    }

    return c;
}

```

## Feedback

Was this editorial helpful?

**Yes**

**No**