Find Missing And Repeating

> 查找数组(1~N)元素中缺失和重复的元素

hash表

```
#include <bits/stdc++.h>

using namespace std;

void findMissRepeat(vector<int> &arr, int N, int &miss, int &repeat) {
    vector<int> hash(N + 1, 0);
    for (int a: arr)
        hash[a]++;
    for (int i = 1; i <= N; ++i) {
        if (hash[i] == 0)
            miss = i;
        if (hash[i] > 1)
            repeat = i;
    }
}

int main() {
    int T;
    scanf("%d\n", &T);
    while (T--) {
        int N, num, miss, repeat;
        vector<int> arr;
        scanf("%d\n", &N);
        for (int i = 0; i < N; ++i) {
            scanf("%d", &num);
            arr.push_back(num);
        }
        findMissRepeat(arr, N, miss, repeat);
        printf("%d %d\n", repeat, miss);
    }
    return 0;
}
```

使用元素作为下标并使用visited标记

对于重复元素：

遍历数组，把 arr[i] - 1 作为下标，将arr[arr[i] - 1] 标记为已访问（设置为负数），如果在遍历接下来的元素时，遇到了 arr[arr[j] - 1] 为负数，则说明arr[j] 该元素重复出现了

对于缺失元素：

在寻找重复元素过程之后，遍历数组，如果arr[i] > 0，则说明 i + 1 在原数组中是没有出现过的

```
#include <bits/stdc++.h>

using namespace std;

void findMissRepeat(vector<int> &arr, int N, int &miss, int &repeat) {
    for (int i = 0; i < N; ++i) {
        if (arr[abs(arr[i]) - 1] > 0)
            arr[abs(arr[i]) - 1] = -arr[abs(arr[i]) - 1];
        else
            repeat = abs(arr[i]);
    }
    for (int i = 0; i < N; ++i) {
        if (arr[i] > 0)
            miss = i + 1;
    }
}

int main() {
    int T;
    scanf("%d\n", &T);
    while (T--) {
        int N, num, miss, repeat;
        vector<int> arr;
        scanf("%d\n", &N);
        for (int i = 0; i < N; ++i) {
            scanf("%d", &num);
            arr.push_back(num);
        }
        findMissRepeat(arr, N, miss, repeat);
        printf("%d %d\n", repeat, miss);
    }
    return 0;
}
```