

C2-F Redraw the Colorless World

题目描述

给定 n 个正整数构成的序列 A_1, A_2, \dots, A_n 以及上界 k ，对于任意 $1 \leq i \leq n$ 有 $1 \leq A_i \leq k$ 。

如果 A 中存在连续的 k 个数恰好包含 $1, 2, \dots, k$ ，则称 A 是多彩的。

你可以执行任意次以下操作：

- 选择不相同的两个下标 i, j ($1 \leq i < j \leq n$)，交换 A_i 与 A_j 。

判断能否将 A 变成多彩的。若能，则求出将 A 变成多彩的所需要的最少操作次数。

输入

本题测试点包含多组数据。

第一行，一个正整数 T ($1 \leq T \leq 1000$)，表示数据组数。

对于每组数据：

第一行，两个正整数 n, k ($1 \leq k \leq n \leq 10^5$)，表示给定正整数序列长度，以及正整数的上界。

第二行， n 个正整数 A_1, A_2, \dots, A_n ($1 \leq A_i \leq k$)，表示给定的正整数。

对于单个测试点，保证有 $\sum n \leq 5 \times 10^5$ 。

输出

对于每组数据：

若能将 A 变成多彩的，则输出一个整数，表示所需的最少操作次数。否则输出一个整数 -1 。

题解

此题为滑动窗口 (sliding window) 的应用。

容易想到，不能将 A 变成多彩的，当且仅当 A 中缺少 $1, 2, \dots, k$ 中的某数。又由 k 为序列的上界，得 A 中出现 $1, 2, \dots, k$ 全部的整数当且仅当 A 中出现不同的数字个数为 k 。因此考虑维护数组 `tcnt` 用于标记某数字是否出现过，统计序列中不同数字的个数，若不为 k 则输出 `-1`。

```
for (int i = 1; i <= n; i++)
{
    cin >> a[i];
    if (!tcnt[a[i]])
    {
        cnt++;
        tcnt[a[i]]++;
    }
}
if (cnt != k)
{
    cout << "-1" << endl;
    return;
}
```

若序列中 $1, 2, \dots, k$ 均出现，则通过有限次交换必能使这些数按某种排列聚集在一起。考虑长度为 k 的子区间，若要使该区间包含 $1, 2, \dots, k$ 的所有数，只需求出区间中已有的不同的数字个数，将缺少的数字与区间外的对应数字交换即可。

注意审题：题目要求 A 中存在连续的 k 个数恰好包含 $1, 2, \dots, k$ ，并非存在一区间的值依次为 $1, 2, \dots, k$ 。

因此，问题转化为求出序列的所有长度为 k 的连续子序列中最大的数字种类数，并用 k 减去该值，即为最少交换次数。

考虑一固定长度 k 的滑动窗口，从左端开始，用 `vcnt` 数组记录当前窗口中各数字出现的次数，用 `cnt` 表示当前窗口中的数字种类数。窗口右端点 `i` 从 1 遍历至 k ，每次遍历后更新数字种类数的最大值 `ans`，最后输出 `k - ans` 即可。

```
#include <cstring>
#include <iostream>
using namespace std;
const int N = 1e5 + 5;
int n, k, cnt, ans;
int a[N], tcnt[N], vcnt[N];
void solve()
{
    cin >> n >> k;
    memset(tcnt + 1, 0, k * sizeof(int));
    memset(vcnt + 1, 0, k * sizeof(int));
    cnt = 0;
    for (int i = 1; i <= n; i++)
    {
        cin >> a[i];
        if (!tcnt[a[i]])
        {
            cnt++;
            tcnt[a[i]]++;
        }
    }
    if (cnt != k)
    {
        cout << "-1" << endl;
        return;
    }
    ans = cnt = 0;
    for (int i = 1; i <= n; i++)
    {
        if (!vcnt[a[i]])
        {
            cnt++;
        }
        vcnt[a[i]]++;
        if (i > k)
        {
            vcnt[a[i - k]]--;
            if (!vcnt[a[i - k]])
            {
                cnt--;
            }
        }
    }
}
```

```

    }
    ans = max(ans, cnt);
}
cout << k - ans << endl;
}

int main()
{
    ios::sync_with_stdio(false);
    int T;
    cin >> T;
    while (T--)
    {
        solve();
    }
    return 0;
}

```

也可以使用 STL 简化 `cnt` 的维护：

```

#include <iostream>
#include <map>
#include <set>
using namespace std;
const int N = 1e5 + 5;
int n, k, cnt, ans, a[N];
void solve()
{
    cin >> n >> k;
    set<int> s;
    map<int, int> m;
    for (int i = 1; i <= n; i++)
    {
        cin >> a[i];
        s.insert(a[i]);
    }
    if ((int)s.size() != k)
    {
        cout << "-1" << endl;
        return;
    }
    ans = 0;
    for (int i = 1; i <= n; i++)
    {
        m[a[i]]++;
        if (i > k)
        {
            m[a[i - k]]--;
            if (m[a[i - k]] == 0)
            {
                m.erase(a[i - k]);
            }
        }
        ans = max(ans, (int)m.size());
    }
}

```

```
        cout << k - ans << endl;
    }

    int main()
    {
        ios::sync_with_stdio(false);
        int T;
        cin >> T;
        while (T--)
        {
            solve();
        }
        return 0;
    }
}
```

相关题目

- [洛谷 P1886 滑动窗口 / 【模板】单调队列](#)
- [力扣 2134. 最少交换次数来组合所有的 1 II](#)
- [力扣 1852. 每个子数组的数字种类数](#)，需要开会员。