

DP问题：最大不连续子列和

例题：1045 Favorite Color Stripe (30分)

Eva is trying to make her own color stripe out of a given one. She would like to keep only her favorite colors in her favorite order by cutting off those unwanted pieces and sewing the remaining parts together to form her favorite color stripe.

It is said that a normal human eye can distinguish about less than 200 different colors, so Eva's favorite colors are limited. However the original stripe could be very long, and Eva would like to have the remaining favorite stripe with the maximum length. So she needs your help to find her the best result.

Note that the solution might not be unique, but you only have to tell her the maximum length. For example, given a stripe of colors {2 2 4 1 5 5 6 3 1 1 5 6}. If Eva's favorite colors are given in her favorite order as {2 3 1 5 6}, then she has 4 possible best solutions {2 2 1 1 1 5 6}, {2 2 1 5 5 5 6}, {2 2 1 5 5 6 6}, and {2 2 3 1 1 5 6}.

Input Specification:

Each input file contains one test case. For each case, the first line contains a positive integer N (≤ 200) which is the total number of colors involved (and hence the colors are numbered from 1 to N). Then the next line starts with a positive integer M (≤ 200) followed by M Eva's favorite color numbers given in her favorite order. Finally the third line starts with a positive integer L (≤ 104) which is the length of the given stripe, followed by L colors on the stripe. All the numbers in a line are separated by a space.

Output Specification:

For each test case, simply print in a line the maximum length of Eva's favorite stripe.

Sample Input:

```
6
5 2 3 1 5 6
12 2 2 4 1 5 5 6 3 1 1 5 6
```

Sample Output:

```
7
```

思路：

1. 这道题给我最大的教训就是.....一定要完完整整完全看懂输出要求.....不然把12和5都当成颜色数据我就一辈子都做不出来QAQ
2. 看清楚题目之后，他就是一道最大不连续序列和，DP问题。

3. 注意需要先踢出不喜欢的颜色的数据。因为 orderTable 中不喜欢的颜色没有被收录进去，他们的顺序值都为0。按照题目的逻辑，由于可以出现重复的数据，使得不喜欢的颜色会被认为是连续的子列。因此需要排除。

参考代码：

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
const int inf = 210;
vector<int> orderTable, stripe, dp;
void init()
{
    int number = 0, index = 0;
    cin >> number;
    orderTable.resize(number+1);
    cin >> number;
    for(int i=1;i<=number;i++)
    {
        cin >> index;
        orderTable[index] = i;
    }
}
int main()
{
    init();
    int temp = 0, maxLen = 0, number = 0;
    cin >> number;
    for (int i = 0; i < number; i++)
    {
        cin >> temp;
        if (orderTable[temp] != 0)
            stripe.push_back(temp);
    }
    dp.assign(stripe.size(), 1);
    for(int i=0;i<stripe.size();i++)
    {
        for(int j=0;j<i;j++)
        {
            if (orderTable[stripe[i]] >= orderTable[stripe[j]])
                dp[i] = max(dp[j] + 1, dp[i]);
        }
        maxLen = max(maxLen, dp[i]);
    }
    cout << maxLen;
    return 0;
}
```