双指针算法

核心思想

```
for (int i = 0, j = 0; i < n; i ++)
{
    while (j < i check(i,j)) j ++;

    // 题目逻辑
}
O(n)
```

```
for (int i = 0; i <n; i ++)
{
    for (int j = 0; j < n;j ++)
    {
        // 题目逻辑
    }
}
O(n2) 将这个算法优化到O(n)
```

例题一

输入一行字符串，将其中每个单词提取出来

输入样例

```
abc def hig
```

输出样例

```
abc
dec
hig
```

代码

```
#include <iostream>
#include <string.h>
using namespace std;

int main()
{
    char str[1000] = "abc def hig";

    int n = strlen(str);
```

```
    for (int i = 0; i < n; i ++)
    {
        int j = i;
        while (j < n && str[j] != ' ')
            j ++;


        // 题目逻辑
        for (int k = i; k < j ; k ++)
            cout<<str[k];
        cout<<endl;

        i = j;

    }

    return 0;
}
```

例题二 最长连续不重复子序列

给定一个长度为 n （n<=10的五次方）的整数序列，请找出最长的不包含重复的数的连续区间，输出它的长度。

输入样例

```
5
1 2 3 5
```

输出样例

```
3
```

代码

```
// 朴素枚举算法
#include <iostream>
#include <set>
using namespace std;
const int N = 1e5+10;


int main()
{
    int a[N], ans = 1, n;

    cin>>n;
    for (int i = 0;i < n; i ++)
        cin>>a[i];


    //枚举做法
    for (int i = 0; i < n; i ++) //头指针
```

```
    {
        for (int j = 0;j <= i; j ++) // 尾指针
        {
            int flag = 0; // 记录[j, i]中是否有重复元素
            set<int> st;
            for (int k = j; k <= i; k ++)
            {
                if (st.count(a[k]))
                {
                    flag = 1;
                    break;
                }
                else
                    st.insert(a[k]);
            }
            if (flag == 0 && i - j + 1 > ans)
                ans = i-j+1;
        }
    }

    cout<<ans;


}
```

```
// 双指针算法
#include <iostream>
using namespace std;
const int N = 1e5+10;

int s[N];

int main()
{
    int a[N], n, ans = 1;
    cin>>n;
    for (int i = 0; i < n; i ++)
        cin>>a[i];

    for (int i = 0, j = 0; i < n; i ++)
    {
        s[a[i]] ++;
        while (s[a[i]] == 2)
        {
            s[a[j]] --;
            j ++;
        }
        if (i - j + 1 > ans)
            ans = i - j + 1;
    }

    cout<<ans;
```

```
    return 0;
}
```

例题三 数组元素的目标和

给定两个升序排序的有序数组 A 和 B，以及一个目标值 x。

数组下标从 0 开始。

请你求出满足 A[i]+B[j]=x 的数对 (i,j)。

数据保证有唯一解。

输入样例

```
4 5 6
1 2 4 7
3 4 6 8 9
```

输出样例

```
1 1
```

代码

```
// 朴素枚举算法
#include <iostream>
using namespace std;
const int N = 1e5+10;

int a[N], b[N];
int n, m, x;


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

    for (int i = 0; i < n; i ++)
        for (int j = 0; j < m; j ++)
            if (a[i] + b[j] == x)
            {
                cout<<i<<" "<<j<<endl;
                return 0;
            }


    return 0;
}
```

```
// 双指针算法
```

```cpp
#include <iostream>
using namespace std;
const int N = 1e5+10;

int a[N], b[N];
int n, m, x;


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

    // i指向a的开头 j 指向 b的结尾  每一次循环j指向第一个使a[i] + b[j] <= x的数
    for (int i = 0, j = m-1; i < n ; i ++)
    {
        while ( j >= 0 && a[i] + b[j] > x) j --;

        if (a[i] + b[j] == x)
        {
            printf("%d %d", i, j);
            break;
        }
    }
    return 0;
}
```

例题四 判断子序列

给定一个长度为 n 的整数序列 a1,a2,…,an以及一个长度为 m 的整数序列 b1,b2,…,bm。

请你判断 a序列是否为 b 序列的子序列。

子序列指序列的一部分项按**原有次序排列**而得的序列，例如序列 {a1,a3,a5}是序列 {a1,a2,a3,a4,a5}的一个子序列。

输入样例

```
3 5
1 3 5
1 2 3 4 5
```

输出样例

```
Yes
```

代码

```cpp
#include <iostream>
using namespace std;
const int N = 1e5+10;


```

```cpp
int n, m;
int a[N], b[N];



int main()
{
    cin>>n>>m;
    for (int i = 0; i < n; i ++)
        scanf("%d", &a[i]);
    for (int i = 0; i < m; i ++)
        scanf("%d", &b[i]);

    for (int i = 0, j = 0;i < n; i ++, j ++)
    {

        while (j < m && a[i] != b[j]) j ++;

        if (j == m)
        {
            cout<<"No"<<endl;
            return 0;
        }
    }
    cout<<"Yes"<<endl;

    return 0;
}
```