

单调栈+单调队列

人员

柳力玮、李瑞涵、刘宸熙、姜皓轩、王馨琪、纪博涵、初锦阳、赵牧之、温郝冬、韩昱辰 到课, 苑钊、刘子轩 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/135>

2025-0629 周日15:30 (综合练习)											
<div>刷新</div>											
#	用户名	姓名	总分	选择分	编程分	时间	A	B	C	D	E
1	liuliwei	柳力玮	500	0	500	2867	100	100	100	100	100
2	hanyuchen	韩昱辰	500	0	500	3922	100	100	100	100	100
3	yuanzhao	苑钊	400	0	400	2023	100	100	100	0	100
4	zhaomuzhi	赵牧之	400	0	400	2639	100	100	100	100	
5	chujinyang	初锦阳	400	0	400	2912	100	100	100	100	
6	liulhan	李瑞涵	400	0	400	4931	100	100	100	100	
7	tianxinyi	田心一	300	0	300	2072	100	100	100		
8	wangxinqi	王馨琪	300	0	300	2387	100	100	100	0	0
9	xiezixuan	谢梓轩	233	0	233	1828	100	100	33		
10	jibohan	纪博涵	200	0	200	2285		100	100		
11	jianghaoxuan	姜皓轩	100	0	100	66				0	100
12	liuchenxi	刘宸熙	90	0	90	0	90				

作业

<https://cppoj.kids123code.com/contest/180> (课上讲了 A ~ B 题, 课后作业是 D E 题, C 题比较难, 同学们可以选择做)

课堂表现

这节课讲的 单调栈 和 单调队列 都比较难一些, 大部分同学们课上都没有完全掌握这两个知识点, 同学们课下可以再多写两遍这两个题加深理解。

课堂内容

突击考试

从 1 到 5 枚举, 计算一个数 最长连续出现了多少次, 最后看 1~5 谁连续出现次数 最大

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

using namespace std;

const int maxn = 100000 + 5;
```

```

int a[maxn], b[maxn], len[10];
int n;

int calc(int t) {
    int cnt = 0, maxx = 0;
    for (int i = 1; i <= n; i++) {
        if (a[i]==t || b[i]==t) cnt++;
        else cnt = 0;
        maxx = max(maxx, cnt);
    }
    return maxx;
}

int main()
{
    cin >> n;
    for (int i = 1; i <= n; i++) cin >> a[i] >> b[i];

    int maxx = 0;
    for (int i = 1; i <= 5; i++) len[i] = calc(i), maxx = max(maxx, len[i]);

    for (int i = 1; i <= 5; i++) {
        if (len[i] == maxx) {
            cout << maxx << " " << i << endl;
            break;
        }
    }
    return 0;
}

```

单调栈

设 $f[i]$ 中, 存的是右边第一个比 $a[i]$ 大的数的下标

从右往左扫, 扫到 $a[i]$ 时, 有意义的数只有一部分, 可以把这一部分有意义的数存放在一个栈中, 这样 $a[i]$ 只需要在这个栈中找右边第一个比他大的即可

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 3e6 + 5;
int w[maxn], f[maxn];

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];

    stack<int> stk;
    for (int i = n; i >= 1; --i) {

```

```
while (!stk.empty() && w[i]>=w[stk.top()]) stk.pop();
f[i] = (stk.empty() ? 0 : stk.top());
stk.push(i);
}

for (int i = 1; i <= n; ++i) cout << f[i] << " ";
cout << endl;
return 0;
}
```

滑动窗口 / 【模板】单调队列

从左往右扫, 每次把一个数从后面放进队列中, 然后把前面的数从队列中删除

新放进去的这个数, 可能会导致从它往前的一部分数变得没有意义了, 从后往前进行删除这一部分无意义的数

每次, 队列中的第一个数就是我们要输出的数

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

using namespace std;

const int maxn = 1e6 + 5;
int w[maxn];

int main()
{
    int n, k; cin >> n >> k;
    for (int i = 1; i <= n; ++i) cin >> w[i];

    deque<int> dq;
    for (int i = 1; i <= n; ++i) {
        if (!dq.empty() && i-dq.front()==k) dq.pop_front();
        while (!dq.empty() && w[i]<=w[dq.back()]) dq.pop_back();
        dq.push_back(i);
        if (i >= k) cout << w[dq.front()] << " ";
    }
    cout << endl;

    dq.clear();
    for (int i = 1; i <= n; ++i) {
        if (!dq.empty() && i-dq.front()==k) dq.pop_front();
        while (!dq.empty() && w[i]>=w[dq.back()]) dq.pop_back();
        dq.push_back(i);
        if (i >= k) cout << w[dq.front()] << " ";
    }
    cout << endl;
    return 0;
}
```