

综合练习

人员

柳力玮、李瑞涵、刘宸熙、王馨琪、初锦阳、赵牧之、韩昱辰、苑钊、田心一、王恩泽、蒋叔璋、刘派、宋云朗、高健桓 到课, 姜皓轩、刘子轩 线上

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/756>

2025-0907 周日15:30 (综合练习)								
<div>刷新</div>								
#	用户名	姓名	编程分	时间	A	B	C	D
1	tianxinyi	田心一	400	2515	100	100	100	100
2	yuanzhao	苑钊	400	4106	100	100	100	100
3	zhaomuzhi	赵牧之	360	1961	100	100	100	60
4	jiangshuzhang	蒋叔璋	300	2064	100	100	100	
5	gaojianhuan	高健桓	300	2171	100	100	100	
6	liuliwei	柳力玮	300	2265	100	100	100	
7	liuchenxi	刘宸熙	300	2418	100	100	100	
8	wangenze	王恩泽	200	1292	100	100		
9	wenhaodong	温郝冬	200	1501	100	100		
10	wangxingqi	王馨琪	200	1526	100	100		
11	hanyuchen	韩昱辰	200	1609	100	100		
12	zouyihang	邹忆航	200	1611	100	100		
13	li Ruihan	李瑞涵	200	1970	100	100		
14	chujinyang	初锦阳	200	2828	100	100	0	
15	liupai	刘派	0	0	0	0		

本周作业

<https://cppoj.kids123code.com/contest/795> (课上讲了 A ~ D 题, 课后作业是 E 题)

课堂表现

今天的 C 题是一个非常经典的二分题目, 这种题目已经做过非常多类似的题目了, 但是大部分同学们课上都做的比较差, 需要老师写一遍之后才能写出来。

这些第一遍没做出来的同学, 课后一定要好好再复习复习这道题, 把这道题完整的掌握。

课堂内容

Dash (上周作业)

set 套 pair 记录哪些位置有可以恢复生命的物品, 然后按照题意模拟走的过程即可

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

using namespace std;
```

```

set<pair<int,int>>s;

int main()
{
    int n, m, h, k; cin >> n >> m >> h >> k;
    string str; cin >> str;
    while (m -- ) { int xx, yy; cin >> xx >> yy; s.insert({xx,yy}); }

    int x = 0, y = 0;
    for (char i : str) {
        if (i == 'R') ++x;
        else if (i == 'L') --x;
        else if (i == 'U') ++y;
        else --y;

        --h;
        if (h < 0) { cout << "No" << endl; return 0; }
        if (s.count({x,y}) && h<k) { h = k; s.erase({x,y}); }
    }
    cout << "Yes" << endl;
    return 0;
}

```

bank

用 set 模拟整个叫号的过程即可

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    set<int> s;
    int n, m; cin >> n >> m;
    int id = 1;

    while (m -- ) {
        int op; cin >> op;
        if (op == 1) {
            s.insert(id); ++id;
        } else if (op == 2) {
            int x; cin >> x; s.erase(x);
        } else {
            cout << *s.begin() << endl;
        }
    }
    return 0;
}

```

Flavors

贪心策略, 最大值一定选, 所以跟最大值类型相同的他们的价值可以直接除 2, 然后选最大的两个即可

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

using namespace std;

const int maxn = 3e5 + 5;
struct node {
    int id, value;
} w[maxn];

bool cmp(node p, node q) {
    return p.value > q.value;
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i].id >> w[i].value;
    sort(w+1, w+n+1, cmp);

    for (int i = 2; i <= n; ++i) {
        if (w[i].id == w[1].id) w[i].value /= 2;
    }
    sort(w+1, w+n+1, cmp);

    cout << w[1].value + w[2].value << endl;
    return 0;
}
```

Minimum Width

二分宽度即可

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

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int w[maxn];

bool check(int n, int m, LL mid) {
    int cnt = 1; LL sum = 0;
    for (int i = 1; i <= n; ++i) {
        if (w[i] > mid) return false;
        if (sum > 0) ++sum;
        if (sum+w[i] <= mid) sum += w[i];
    }
    return cnt >= 2;
}
```

```

        else ++cnt, sum = w[i];
    }
    return cnt <= m;
}

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

    LL l = 1, r = 2e14+5;
    while (l <= r) {
        LL mid = (l + r) / 2;
        if (check(n, m, mid)) r = mid-1;
        else l = mid+1;
    }
    cout << l << endl;
    return 0;
}

```

Prefix K-th Max

用小根堆 维护前 k 大的数

如果一个数大于小根堆里的最小值, 就把最小值删除, 把这个数插入小根堆中

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n, k; cin >> n >> k;
    priority_queue<int, vector<int>, greater<int>>>q;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x;
        if (i <= k) {
            q.push(x);
            if (i == k) cout << q.top() << endl;
        }
        else {
            if (x > q.top()) {
                q.pop(); q.push(x);
            }
            cout << q.top() << endl;
        }
    }
    return 0;
}

```