

# 综合练习

## 人员

王毅博、阮文璋、褚锦轩、王承周、许睿谦 到课

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1115>



The screenshot shows a competition results page with a table of scores for seven participants. The table has columns for #, 用户名 (Username), 姓名 (Name), 编程分 (Programming Score), 时间 (Time), A, B, C, and D. The scores for all problems (A, B, C, D) are 100 for all participants except the last one who scored 91.

#	用户名	姓名	编程分	时间	A	B	C	D
1	wangyibo	王毅博	391	863	100	100	100	91
2	ruanwenzhang	阮文璋	300	352	100	100	100	
3	chujinxuan	褚锦轩	300	607	100	100	100	
4	wangchengzhou	王承周	300	1614	100	100	100	
5	chenxinmiao	陈欣妙	200	338	100	100		
6	dongyuhan	董昱含	200	1011	100	100		
7	xuruiqian	许睿谦	100	188	100			

## 本周作业

<https://cppoj.kids123code.com/contest/1210> (课上讲了 A ~ C 题, 课后作业是 D 题)

## 课堂表现

今天上课同学们听讲做题都比较认真, 希望同学们继续保持。

## 课堂内容

### 良好的感觉 (上周作业)

枚举最小值 i, 找最左和最右, 跟之前的 [蓝桥杯 2023 国 C] 最大区间 这道题基本一样

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

using namespace std;

typedef long long LL;
const int N = 3e5 + 5, M = 20;
int w[N], f[N][M], _lg2[N];
LL p[N];
int n;

LL get_sum(int l, int r) { return p[r] - p[l-1]; }
```

```
int get_min(int l, int r) {
    int len = r - l + 1;
    int k = _lg2[len];
    return min(f[l][k], f[r-(1<<k)+1][k]);
}

int find_L(int id) {
    int l = 1, r = id;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (get_min(mid, id) == w[id]) r = mid-1;
        else l = mid+1;
    }
    return l;
}

int find_R(int id) {
    int l = id, r = n;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (get_min(id, mid) == w[id]) l = mid+1;
        else r = mid-1;
    }
    return r;
}

LL calc(int id) {
    int l = find_L(id), r = find_R(id);
    return (LL)w[id] * get_sum(l, r);
}

int main()
{
    for (int i = 0; (1<<i) < N; ++i) _lg2[1<<i] = i;
    for (int i = 1; i < N; ++i) {
        if (!_lg2[i]) _lg2[i] = _lg2[i-1];
    }

    cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i], f[i][0] = w[i], p[i] = p[i-1] + w[i];
    for (int k = 1; k < M; ++k) {
        for (int i = 1; i+(1<<k)-1 <= n; ++i) {
            f[i][k] = min(f[i][k-1], f[i+(1<<(k-1))][k-1]);
        }
    }

    LL res = 0;
    for (int i = 1; i <= n; ++i) res = max(res, calc(i));
    cout << res << endl;
    return 0;
}
```

## Election Quick Report

set 里存放每个候选人编号以及候选人的得票数

每次给一个人加票时, 从 set 里删掉这个人的信息, 并重新插入新的信息即可

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

using namespace std;

const int maxn = 200000 + 5;
int f[maxn];
struct node {
    int id, cnt;
    bool operator < (const node& p) const {
        if (cnt != p.cnt) return cnt > p.cnt;
        return id < p.id;
    }
};

int main()
{
    set<node> s;
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) s.insert({i, 0});
    while (m -- ) {
        int x; cin >> x;
        int t = f[x]; ++f[x];
        auto it = s.find({x, t});
        s.erase(it); s.insert({x, t+1});
        cout << (*s.begin()).id << endl;
    }
    return 0;
}
```

## Take ABC

类似于括号匹配, 用栈存未删除的字符, 当这个字符是C, 且前两个是B和A的时候, 删除; 否则, 把字符往栈里放

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

using namespace std;

const int maxn = 2e5 + 5;
char s[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);
```

```

vector<char> vec; int len = 0;
for (int i = 1; i <= n; ++i) {
    if (s[i]=='C' && len>=2 && vec[len-1]=='B' && vec[len-2]=='A') {
        vec.pop_back(), vec.pop_back(), len -= 2;
    } else vec.push_back(s[i]), ++len;
}

for (char i : vec) cout << i;
cout << endl;
return 0;
}

```

## [CSP-J 2022] 上升点列

$dp, f[i][j]$ : 考虑以第  $i$  个点结尾, 前面一共添加了  $j$  个点时, 能得到的序列的最大长度是多少

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 500 + 5;
struct node {
    int x, y;
    bool operator < (const node& p) const {
        if (x != p.x) return x < p.x;
        return y < p.y;
    }
} w[maxn];
int f[maxn][maxn];

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

    for (int i = 1; i <= n; ++i) {
        for (int j = 0; j <= m; ++j) {
            f[i][j] = j+1;
            for (int k = 1; k < i; ++k) {
                if (w[i].y < w[k].y) continue;

                int c = w[i].x-w[k].x + w[i].y-w[k].y - 1;
                if (j-c >= 0) f[i][j] = max(f[i][j], f[k][j-c]+c+1);
            }
        }
    }

    int res = 0;
    for (int i = 1; i <= n; ++i) res = max(res, f[i][m]);
}

```

```
cout << res << endl;
return 0;
}
```