

综合混练

人员

秦显森、刘闯速、赵熙羽、牛同泽、高健桓、陈洛冉、程梓豪、谢亚锴、隋钰涵 到课, 孙靖轲、于子珈、郭骐嘉 线上

上周作业检查

https://www.luogu.com.cn/contest/248738

2025-0525周日10:30

报名

编辑比赛

题目数5 | 报名人数15

比赛说明 | 题目列表 | 排行榜

名次	参赛者	总分	A	B	C	D	E
#1	程梓豪	420 (7.03d)	100 (2.30s)	100 (7ms)	20 (0ms)	100 (2.16h)	100 (6.94d)
#2	孙靖轲	420 (9.86d)	100 (3.81s)	100 (7ms)	20 (4.42d)	100 (2.01s)	100 (5.44d)
#3	赵熙羽	400 (5.94d)	100 (3.85s)	100 (7ms)	0 (0ms)	100 (2.00s)	100 (5.94d)
#4	陈洛冉	320 (4.38s)	100 (2.34s)	100 (7ms)	20 (0ms)	100 (2.03s)	0
#5	于子珈	320 (5.83s)	100 (3.83s)	100 (8ms)	20 (0ms)	100 (2.00s)	0
#6	谢亚锴	320 (5.84s)	100 (3.83s)	100 (7ms)	20 (0ms)	100 (2.00s)	
#7	秦显森	320 (2.03h)	100 (2.34s)	100 (8ms)	20 (2.03h)	100 (791ms)	
#8	高健桓	320 (2.81d)	100 (3.87s)	100 (6ms)	20 (1.40d)	100 (1.41d)	
#9	隋钰涵	320 (6.92d)	100 (3.85s)	100 (6ms)	20 (6.92d)	100 (743ms)	
#10	牛同泽	320 (7.07d)	100 (2.46s)	100 (7ms)	20 (6.98d)	100 (2.07h)	
#11	郭骐嘉	300 (4.71s)	100 (3.85s)	100 (8ms)		100 (855ms)	
#12	杨谨硕	300 (5.79s)	100 (3.83s)	100 (7ms)		100 (1.96s)	
#13	刘闯速	210 (2.35s)	100 (2.34s)	100 (7ms)		10 (0ms)	
#14	aiyishengaiyishi	210 (2.03h)	100 (3.82s)	100 (7ms)	0 (0ms)	10 (2.03h)	

作业

https://www.luogu.com.cn/contest/250020 (课上讲了 A ~ D 题, 课后作业是 E 题)

课堂表现

今天课上的 A 题满分做法比较复杂一些, 许多同学课上在老师讲解下过了这道题, 但不一定是真的掌握了, 课下需要再好好复习复习这道题。

课堂内容

P9426 [蓝桥杯 2023 国 B] 抓娃娃

关键点: 每次询问的区间长度, 都会 \geq 每一个线段的长度

看一个区间是否覆盖了一半的线段, 就看这个区间是否覆盖了一个线段的中点即可

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

using namespace std;

const int maxn = 2e6 + 5;
int w[maxn], p[maxn];

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

int main()
{
    int n, T; cin >> n >> T;
    while (n -- ) {
        int l, r; cin >> l >> r; w[l+r]++;
    }
    for (int i = 1; i < maxn; ++i) p[i] = p[i-1] + w[i];

    while (T -- ) {
        int l, r; cin >> l >> r;
        cout << get_sum(2*l, 2*r) << endl;
    }
    return 0;
}
```

U480698 Wandering

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

using namespace std;

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

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];
    for (int i = 1; i <= n; ++i) {
        p[i] = p[i-1] + w[i];
        pMax[i] = max(pMax[i-1], p[i]);
    }
}
```

```

}

LL pos = 0, res = 0;
for (int i = 1; i <= n; ++i) {
    res = max(res, pos + pMax[i]);
    pos += p[i];
}
cout << res << endl;
return 0;
}

```

P12141 [蓝桥杯 2025 省 A] 红黑树

第 n 行第 k 个点的颜色, 跟第 $n-1$ 行第 $(k+1)/2$ 个点的颜色有关

进行递归搜索即可

```

#include <bits/stdc++.h>

using namespace std;

int dfs(int n, int k) { // 求第 n 行的第 k 列
    if (n==1 && k==1) return 1;
    if (n == 2) {
        if (k == 1) return 1;
        else return 0;
    }

    if (k%2 == 0) return 1 - dfs(n-1, (k+1)/2);
    else return dfs(n-1, (k+1)/2);
}

void solve() {
    int n, k; cin >> n >> k;
    int value = dfs(n, k);
    if (value == 0) cout << "BLACK" << endl;
    else cout << "RED" << endl;
}

int main()
{
    int T; cin >> T;
    while (T -- ) solve();
    return 0;
}

```

P2207 Photo

输入 k 组关系, 每组是 l, r 关系不好, 所以我们应该在 $[l, r-1]$ 区间中挑一个点画一条隔断线

那么原问题的多组关系, 现在就转变为了: 有多个区间, 我们要选尽可能少的点, 保证每个区间内都至少包含一个点

做法跟 凌乱的yyy 这道题一致, 按右端点排序, 然后从左往右扫一遍即可

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

using namespace std;

const int maxn = 1000 + 5;
struct node {
    int l, r;
    bool operator < (const node& p) const { return r < p.r; }
} w[maxn];

int main()
{
    int limit, n; cin >> limit >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> w[i].l >> w[i].r;
        if (w[i].l > w[i].r) swap(w[i].l, w[i].r);
        --w[i].r;
    }
    sort(w+1, w+n+1);

    int res = 0, rmax = -1;
    for (int i = 1; i <= n; ++i) {
        if (w[i].l > rmax) ++res, rmax = w[i].r;
    }
    cout << res+1 << endl;
    return 0;
}
```

P1803 凌乱的yyy / 线段覆盖

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

using namespace std;

const int maxn = 1e6 + 5;
struct node {
    int l, r;
} w[maxn];

bool cmp(node p, node q) {
    return p.r < q.r;
}

int main()
{

```

```
int n; cin >> n;
for (int i = 1; i <= n; ++i) cin >> w[i].l >> w[i].r;
sort(w+1, w+n+1, cmp);

int last = -1, res = 0;
for (int i = 1; i <= n; ++i) {
    if (w[i].l >= last) {
        ++res;
        last = w[i].r;
    }
}

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