

综合混练

人员

李雨谦、杨俊彦、龙沛轩、李锦澍、袁晨峻、徐思远、周治润 到课, 杨咏丞 线上

上周作业检查

https://www.luogu.com.cn/contest/248672

2025-0525六队上课(位运算)

报名

编辑比赛

题目数6 | 报名人数21

比赛说明 | 题目列表 | 排行榜

名次	参赛者	总分	A	B	C	D	E	F
#1	杨俊彦	520 (2.02h)	100 (3.82s)	100 (7ms)	20 (0ms)	100 (738ms)	100 (2.04s)	100 (2.01h)
#2	徐思远	520 (5.28h)	100 (2.32s)	100 (7ms)	20 (0ms)	100 (2.12s)	100 (1.95h)	100 (3.33h)
#3	阮文章	520 (1.23d)	100 (3.79s)	100 (7ms)	20 (13.11h)	100 (872ms)	100 (2.36h)	100 (14.06h)
#4	刘锦轩	520 (4.23d)	100 (2.33s)	100 (7ms)	20 (13.82h)	100 (13.08h)	100 (1.56d)	100 (1.56d)
#5	袁晨峻	520 (5.46d)	100 (716ms)	100 (7ms)	20 (0ms)	100 (296ms)	100 (401ms)	100 (5.46d)
#6	王承周	520 (11.66d)	100 (2.32s)	100 (7ms)	20 (3.45d)	100 (6.95h)	100 (3.46d)	100 (4.46d)
#7	董昱含	520 (13.80d)	100 (7.76h)	100 (7.98h)	20 (5.50d)	100 (13.27h)	100 (3.56d)	100 (3.53d)
#8	刘奕辰	520 (15.64d)	100 (2.32s)	100 (7ms)	20 (6.56d)	100 (539ms)	100 (2.52d)	100 (6.57d)
#9	李锦澍	500 (19.35d)	100 (3.83s)	100 (7ms)		100 (6.44d)	100 (6.44d)	100 (6.47d)
#10	陈欣妙	420 (1.95h)	100 (3.84s)	100 (7ms)	20 (0ms)	100 (823ms)	100 (1.95h)	
#11	许睿谦	420 (1.07d)	100 (3.82s)	100 (7ms)	20 (12.69h)	100 (662ms)	100 (12.93h)	
#12	杨咏丞	420 (6.96d)	100 (3.84s)	100 (7ms)	20 (3.47d)	100 (2.10s)	100 (3.50d)	
#13	龙沛轩	330 (6.99d)	100 (2.50s)	100 (8ms)	20 (0ms)	10 (0ms)		100 (6.99d)
#14	曹嫻	320 (12.55h)	100 (3.82s)	100 (7ms)	20 (0ms)		100 (12.55h)	
#15	SSJ司云心	320 (5.17d)	100 (3.83s)	100 (7ms)	20 (1.59d)	100 (3.58d)	0	
#16	王毅博	300 (3.00s)	100 (2.33s)	100 (7ms)	0 (0ms)	100 (662ms)		
#17	褚锦轩	300 (3.01s)	100 (2.34s)	100 (7ms)	0 (0ms)	100 (664ms)		
#18	周治润	300 (4.69s)	100 (3.85s)	100 (7ms)	0 (0ms)	100 (831ms)	0 (0ms)	
#19	李雨谦	210 (2.32s)	100 (2.31s)	100 (7ms)		10 (0ms)		
#20	王陆文龙	200 (3.79s)	100 (3.78s)	100 (7ms)	0	0 (0ms)		

作业

<https://www.luogu.com.cn/contest/249994> (课上讲了 A ~ D 题, 课后作业是 E 题)

课堂表现

今天课上给同学们出了 2 道 dp 题, 同学们课上做起来有点吃力, 课后要再好好复习复习这 2 道题。

课堂内容

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;
}
```

P1934 封印

$f[i]$: 1- \rightarrow i 最小花费

答案: $f[n]$

从左往右求 $f[i]$

```
for (int i = 1; i <= n; i++) {
```

```

    求 f[i] ?
    枚举 i 前面的 j: 先从1到j-1花费最小, 再从j到i
}

```

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 10000 + 5;
int w[maxn], p[maxn];
LL f[maxn];

int get_sum(int l, int r) { return (l<=r ? p[r]-p[l-1] : 0); }

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

    memset(f, 0x3f, sizeof(f));
    f[0] = 0;
    for (int i = 1; i <= n; ++i) {
        f[i] = f[i-1] + (LL)w[i]*n*n;
        for (int j = 1; j < i; j++) {
            if (w[j]+w[i] <= t) f[i] = min(f[i], f[j-1] + ((LL)w[j]+w[i])*get_sum(j,i));
        }
    }
    cout << f[n] << endl;
    return 0;
}

```

P2193 HXY和序列

$f[i][j]$: 以第 i 个数结尾, 第 i 个数填 j 时, 一共有多少方案

$f[i][j]$ 可以由所有的 $f[i-1][k]$ 转移而来 (k 需要是 i 的因数)

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e3 + 5;
const int mod = 1e9 + 7;
int f[maxn][maxn];
vector<int> vec[maxn];

int main()

```

```

{
    for (int i = 1; i < maxn; ++i) {
        for (int j = 1; j <= i; ++j) {
            if (i%j == 0) vec[i].push_back(j);
        }
    }

    int limit, n; cin >> limit >> n;

    f[0][1] = 1;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= limit; ++j) {
            for (int k : vec[j]) {
                if (j%k == 0) f[i][j] = (f[i][j] + f[i-1][k]) % mod;
            }
        }
    }

    int res = 0;
    for (int i = 1; i <= limit; ++i) res = (res + f[n][i]) % mod;
    cout << res << endl;
    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;
}
```