

综合练习

人员

杨俊彦、徐思远、刘奕辰、李雨谦、陈欣妙 到课, 杨咏丞、周治润 线上

上周作业检查

https://www.luogu.com.cn/contest/252013

2025-06-15 六年级上 课(开营课)

报名

编辑比赛

题目数6 | 报名人数18

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 | 徐思远 | 600 (1.93h) | 100 (2.10s) | 100 (38ms) | 100 (202ms) | 100 (35ms) | 100 (37ms) | 100 (1.93h) |
| #2 | 杨俊彦 | 600 (7.39h) | 100 (9.72s) | 100 (59ms) | 100 (573ms) | 100 (2.16h) | 100 (2.75h) | 100 (2.48h) |
| #3 | 阮文璋 | 600 (10.22d) | 100 (9.08s) | 100 (55ms) | 100 (514ms) | 100 (3.05d) | 100 (4.11d) | 100 (3.06d) |
| #4 | 陈欣妙 | 600 (30.66d) | 100 (5.09d) | 100 (5.09d) | 100 (5.10d) | 100 (5.11d) | 100 (5.13d) | 100 (5.14d) |
| #5 | SSJ司云心 | 500 (8.02d) | 100 (8.48s) | 100 (55ms) | 100 (528ms) | 100 (4.48d) | | 100 (3.54d) |
| #6 | 王承周 | 500 (10.21d) | 100 (8.19s) | 100 (56ms) | 100 (526ms) | 100 (5.10d) | | 100 (5.11d) |
| #7 | 刘奕辰 | 500 (12.69d) | 100 (9.20s) | 100 (56ms) | 100 (525ms) | 100 (6.09d) | | 100 (6.59d) |
| #8 | 李雨谦 | 500 (35.39d) | 100 (7.08d) | 100 (7.08d) | 100 (7.08d) | 100 (7.08d) | | 100 (7.08d) |
| #9 | 潘俊伊 | 400 (10.05s) | 100 (9.44s) | 100 (55ms) | 100 (524ms) | 100 (37ms) | | |
| #10 | 杨咏丞 | 400 (6.46d) | 100 (9.21s) | 100 (56ms) | 100 (527ms) | 100 (6.46d) | | |
| #11 | 褚锦轩 | 400 (7.13d) | 100 (10.60s) | 100 (58ms) | 100 (577ms) | 100 (7.13d) | | |
| #12 | 许睿谦 | 300 (4.56s) | 100 (4.30s) | 100 (50ms) | 100 (217ms) | | | |
| #13 | 王毅博 | 300 (4.71s) | 100 (4.45s) | 100 (47ms) | 100 (214ms) | | | 0 |
| #14 | 周治润 | 300 (6.40d) | 100 (9.08s) | 100 (53ms) | 100 (6.40d) | | | |
| #15 | 徐易 | 100 (5.14d) | | | | | | 100 (5.14d) |

作业

https://cppoj.kids123code.com/contest/105 (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

今天的 B 题比较复杂一些, 需要用到 二分套 比较复杂的前缀和, 同学们课上整体做的不是很好, 课后一定要好好复习一下这道题。

课堂内容

P2814 家谱

并查集裸题, 多一步 要把字符串映射为整数, 把整数映射为字符串 的过程

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

using namespace std;

const int maxn = 5e4 + 5;
int f[maxn];

int fFind(int x) {
    if (f[x] != x) f[x] = fFind(f[x]);
    return f[x];
}

string str[maxn];

int main()
{
    int n = 0, id = 0;
    map<string, int> mp;
    map<int, string> mp2;
    while (true) {
        ++n;
        cin >> str[n];
        if (str[n] == "$") break;
        string s = str[n].substr(1);
        if (!mp.count(s)) {
            ++id; mp[s] = id; mp2[id] = s;
        }
    }

    for (int i = 1; i <= n-1; ++i) f[i] = i;

    int fa_id = 0;
    for (int i = 1; i <= n-1; ++i) {
        int id = mp[str[i].substr(1)];
        if (str[i][0] == '#') fa_id = id;
        else if (str[i][0] == '+') {
            int f1 = fFind(id), f2 = fFind(fa_id);
            f[f1] = f2;
        }
        else {
            int t = fFind(id);
            cout << mp2[id] << " " << mp2[t] << endl;
        }
    }
}
```

```

    }
    return 0;
}

```

[蓝桥杯 2021 省 AB] 砝码称重

0/1 背包的 dp, 考虑一个砝码选或不选

选砝码的话, 可以考虑 加上这个砝码/用前面减这个砝码/用这个砝码减前面 这 3 种情况

```

1  1. 定义状态:
2      f[i][j]: 看前 i 个砝码, 能否凑出来 j 这个值
3      如果 f[i][j]==true, 说明能凑出来
4      |      f[i][j]==false, 说明凑不出来
5
6  2. 如何通过状态求答案:
7      f[n][1]==true
8      f[n][2]==true
9      f[n][3]==true
10     ...
11     f[n][1e5]==true
12
13 3. 状态转移:
14  f[i][j] = {
15     |      1. 第 i 个砝码不用, f[i-1][j]
16     |      2. 第 i 个砝码用:
17     |          f[i-1][j-a[i]], f[i-1][j+a[i]], f[i-1][a[i]-j]
18     |      }
19
20 4. 初值定义:
21     f[0][0] = true / f[1][a[1]] = true
22

```

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 5;
bool f[105][maxn];

int main()
{
    f[0][0] = true;
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x;
        for (int j = 0; j < maxn; ++j) {

```

```

        f[i][j] |= f[i-1][j];
        if (j+x < maxn) f[i][j+x] |= f[i-1][j];
        if (j-x >= 0) f[i][j-x] |= f[i-1][j];
        if (x-j >= 0) f[i][x-j] |= f[i-1][j];
    }
}

int res = 0;
for (int i = 1; i < maxn; ++i) {
    if (f[n][i]) ++res;
}
cout << res << endl;
return 0;
}

```

[蓝桥杯 2021 国 ABC] 123

首先用二分确定一个数在 第几行第几列, 然后可以利用前缀和可以 $O(1)$ 计算前面这一部分的和

```

1  1. l, r:
2      calc(x): 第一个数 到 第x个数的和
3      calc(r) - calc(l-1)
4
5  2. calc(x):
6      首先要先确定 x 的位置, 假设 x 在 a行b列 这个位置
7      要确定 x 在第几行, 可以通过 二分 来求行
8      求完第 a 行之后, 如何确定列?
9      b = x - (c[1]+c[2]+...+c[a-1])
10
11 3. 前 a-1 行的和求出来, 1+2+3+...+b
12     c[1] = 1
13     c[2] = 1+2
14     c[3] = 1+2+3
15     ...
16
17     前 a-1 行就是 c[1]+c[2]+...+c[a-1]

```

```

#include <bits/stdc++.h>

using namespace std;

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

LL g_value(int l, int r) { return ((LL)l+r)*(r-l+1)/2; }

```

```

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

int get_row(LL x) {
    int l = 1, r = 2e6;
    while (l <= r) {
        int mid = (l + r) / 2;
        if (g_value(1,mid) >= x) r = mid-1;
        else l = mid+1;
    }
    return l;
}

LL calc(LL x) {
    int a = get_row(x);
    int b = x - g_value(1, a-1);
    return get_sum(1, a-1) + g_value(1, b);
}

void solve() {
    LL ll, rr; cin >> ll >> rr;
    cout << calc(rr) - calc(ll-1) << endl;
}

int main()
{
    for (int i = 1; i < maxn; ++i) w[i] = g_value(1, i), p[i] = p[i-1] + w[i];
    int T; cin >> T;
    while (T -- ) solve();
    return 0;
}

```

最后的迷宫

把终点向外延伸的 8 个方向能访问到的地方都打上标记, 然后从起点跑 bfs, 看最快什么时候能跑到标记点

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 16384 + 5;
vector<char> vec[maxn];
vector<int> dis[maxn];
vector<bool> st[maxn];
int n, m;
struct node {
    int x, y;
};
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};
int dx2[] = {-1, -1, -1, 0, 0, 1, 1, 1};
int dy2[] = {-1, 0, 1, -1, 1, -1, 0, 1};

```

```
int bfs(int sx, int sy, int ex, int ey) {
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) dis[i][j] = -1, st[i][j] = false;
    }

    for (int i = 0; i < 8; ++i) {
        int x = ex, y = ey;
        while (x>=1 && x<=n && y>=1 && y<=m && vec[x][y]=='0') {
            st[x][y] = true;
            x += dx2[i], y += dy2[i];
        }
    }

    queue<node> q; q.push({sx, sy}); dis[sx][sy] = 0;
    while (!q.empty()) {
        node u = q.front(); q.pop();
        int x = u.x, y = u.y;
        if (st[x][y]) return dis[x][y];
        for (int i = 0; i < 4; ++i) {
            int nx = x+dx[i], ny = y+dy[i];
            if (nx>=1 && nx<=n && ny>=1 && ny<=m && vec[nx][ny]=='0' && dis[nx][ny]==-1)
            {
                q.push({nx, ny}); dis[nx][ny] = dis[x][y] + 1;
            }
        }
    }

    return -1;
}

int main()
{
    cin >> n >> m;
    for (int i = 0; i <= n+2; ++i) {
        for (int j = 0; j <= m+2; ++j) {
            vec[i].push_back(' ');
            dis[i].push_back(0);
            st[i].push_back(false);
        }
    }

    for (int i = 1; i <= n; ++i) {
        string str; cin >> str; str = " " + str;
        for (int j = 1; j <= m; ++j) vec[i][j] = str[j];
    }

    while (true) {
        int sx, sy, ex, ey; cin >> ex >> ey >> sx >> sy;
        if (!sx) break;
        int t = bfs(sx, sy, ex, ey);
        if (t == -1) cout << "Poor Harry" << endl;
        else cout << t << endl;
    }
}
```

```
    return 0;  
}
```