

综合混练

人员

杨瑾硕、秦显森、刘闯速、孙靖轲、赵熙羽、牛同泽、于子珈、高健桓、陈洛冉、董浩桢、牟茗、程梓豪 到课，郭骐嘉、谢亚锴 线上

上周作业检查

<https://www.luogu.com.cn/contest/246119>

2025-0511周日10:30

报名

编辑比赛

题目数4 | 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
#1	陈洛冉	389 (7.39d)	100 (27ms)	100 (77ms)	89 (8.09h)	100 (7.06d)
#2	孙靖轲	388 (11.80d)	100 (22ms)	100 (1.39d)	100 (4.41d)	88 (6.00d)
#3	谢亚锴	388 (20.11d)	100 (26ms)	100 (6.53d)	100 (6.58d)	88 (7.00d)
#4	郭骐嘉	300 (12.94d)	100 (23ms)	100 (6.45d)	100 (6.49d)	
#5	高健桓	250 (13.88d)	100 (24ms)	50 (6.83d)	100 (7.05d)	
#6	秦显森	240 (8.58d)	100 (680ms)	40 (4.29d)	100 (4.29d)	0
#7	程梓豪	227 (12.32h)	100 (3.70h)	100 (3.92h)	27 (4.70h)	
#8	于子珈	213 (7.31d)	100 (29ms)	0 (0ms)	13 (6.17h)	100 (7.05d)
#9	赵熙羽	200 (6.46d)	100 (32ms)	100 (6.46d)		
#10	牛同泽	200 (6.89d)	100 (23ms)	100 (6.89d)		
#11	隋钰涵	140 (527ms)	100 (527ms)	40 (0ms)		
#12	刘闯速	140 (2.06h)	100 (27ms)	40 (2.06h)		
#13	庞瀉	100 (29ms)	100 (29ms)			
#14	董浩桢	100 (30ms)	100 (30ms)			
#15	杨瑾硕	100 (481ms)	100 (481ms)			

作业

<https://www.luogu.com.cn/contest/247006> (课上讲了 A ~ C 题, 课后作业是 D 题)

课堂表现

同学们课上听讲都很认真, 今天的 A 题和 C 题, 这两道题同学们课上普遍做的不是太好, 课下要好好复习再做一做。

课堂内容

P11272 「Diligent-OI R1 B」 DlgArray

1. 当 $k > r-l$ 时, 无解
2. 当 $k == r-l$ 时, $[l,r]$ 这个区间里可以是 全1 或者是 $r-l$ 个1
3. 当 $k < r-l$ 时, $[l,r]$ 这个区间里必须是 k 个1

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

using namespace std;

const int maxn = 1e6 + 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;
    for (int i = 1; i <= n; ++i) scanf("%d",&w[i]), p[i] = p[i-1] + w[i];
    while (T -- ) {
        int l, r, k; scanf("%d%d%d",&l,&r,&k);
        if (k > r-l) cout << -1 << "\n";
        else {
            if (k==r-l && get_sum(l,r)==r-l+1) cout << 0 << "\n";
            else cout << abs(k-get_sum(l,r)) << "\n";
        }
    }
    return 0;
}
```

U537160 move

bfs, 把每个字母对应的位置存下来, 一个字母只会做一次传送操作

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

using namespace std;

const int maxn = 2000 + 5;
char s[maxn][maxn];
int dis[maxn][maxn];

int get_value(char x) { return x-'a'+1; }
bool f[30];
struct node {
    int x, y;
```

```

};
vector<node> vec[30];
int n, m, sx, sy, ex, ey;
int dx[] = {-1, 1, 0, 0}, dy[] = {0, 0, -1, 1};

void bfs() {
    memset(dis, -1, sizeof(dis));
    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;
        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 && s[nx][ny]!='#' && dis[nx][ny]==-1) {
                q.push({nx,ny}); dis[nx][ny] = dis[x][y]+1;
            }
        }

        if (islower(s[x][y]) && !f[get_value(s[x][y])]) {
            int t = get_value(s[x][y]); f[t] = true;
            for (node it : vec[t]) {
                int nx = it.x, ny = it.y;
                if (dis[nx][ny] == -1) q.push({nx,ny}), dis[nx][ny] = dis[x][y]+1;
            }
        }
    }
}

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

    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            if (s[i][j] == 'S') sx = i, sy = j;
            if (s[i][j] == 'G') ex = i, ey = j;
            if (islower(s[i][j])) vec[get_value(s[i][j])].push_back({i, j});
        }
    }

    bfs();

    cout << dis[ex][ey] << endl;
    return 0;
}

```

P3056 [USACO12NOV] Clumsy Cows S

栈模拟左右括号匹配

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 1e5 + 5;
char s[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1), res = 0;
    stack<char> stk;
    for (int i = 1; i <= n; ++i) {
        char x = s[i];
        if (x == '(') stk.push(x);
        else {
            if (!stk.empty()) stk.pop();
            else {
                stk.push('('), ++res;
            }
        }
    }
    res += stk.size()/2;
    cout << res << endl;
    return 0;
}

```

U537158 floor

记忆化搜索, 维护 $f[x][y][id]$ 的信息, 代表是否到过 (x,y) 这个点, 沿着 id 方向 这个状态

那么 $dfs(x,y,id)$ 的下一个状态, 则就是看下一个位置 $(x+dx[id],y+dy[id])$ 是 '.' 还是 '#' 即可

如果是 '.', 下一个状态就是 $dfs(x+dx[id], y+dy[id], id)$;

如果是 '#', 下四个状态就是 $dfs(x, y, 0/1/2/3)$;

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 200 + 5;
char s[maxn][maxn];
bool f[maxn][maxn][4];
int dx[] = {0, 0, -1, 1}, dy[] = {-1, 1, 0, 0};
int n, m;

void dfs(int x, int y, int id) {
    if (f[x][y][id]) return;
    f[x][y][id] = true;
}

```

```
int nx = x+dx[id], ny = y+dy[id];
if (s[nx][ny] == '.') return dfs(nx, ny, id);

for (int i = 0; i < 4; ++i) dfs(x, y, i);
}

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

    for (int i = 0; i < 4; ++i) dfs(2, 2, i);

    int res = 0;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            int cnt = 0;
            for (int k = 0; k < 4; ++k) {
                if (f[i][j][k] && s[i][j]=='.') cnt = 1;
            }
            res += cnt;
        }
    }
    cout << res << endl;
    return 0;
}
```