

CSP T2 选讲

张俊焕

2025 年 9 月 8 日

HUST

- CSP T2 是 CSP 五题中的第二题

- CSP T2 是 CSP 五题中的第二题
- CSP 的五题一般是语法题 + 语法（暴力）题 + 大模拟题 + 稍复杂算法题（图论，数论，dp 等） + 数据结构

- CSP T2 是 CSP 五题中的第二题
- CSP 的五题一般是语法题 + 语法（暴力）题 + 大模拟题 + 稍复杂算法题（图论，数论，dp 等） + 数据结构
- 想要拿到 300+ 的一般要 AC 前两题，大模拟尽量拿高分，剩下两题写暴力。

- 题面

第二届. 画图

- 题面
- 将格子用其右上角坐标表示

- 题面
- 将格子用其右上角坐标表示
- 我们发现操作等价于给 $(X_1 + 1, Y_1 + 1) \dots (X_2, Y_2)$ 矩形中的格子涂色

第二届. 画图

- 题面
- 将格子用其右上角坐标表示
- 我们发现操作等价于给 $(X_1 + 1, Y_1 + 1) \dots (X_2, Y_2)$ 矩形中的格子涂色
- 由于数据范围很小，可以每次暴力覆盖，用一个数组记录每个点是否被涂色

第二届. 画图

- 题面
- 将格子用其右上角坐标表示
- 我们发现操作等价于给 $(X_1 + 1, Y_1 + 1) \dots (X_2, Y_2)$ 矩形中的格子涂色
- 由于数据范围很小，可以每次暴力覆盖，用一个数组记录每个点是否被涂色
- 最后统计答案

第二届. 画图

```
#include <bits/stdc++.h>
#define For(i, l, r) for (int i=l; i<=r; ++i)
using namespace std;

const int N = 105;
int n, L1, L2, R1, R2, a[N][N];
signed main() {
    cin >> n;
    For(i, 1, n) {
        cin >> L1 >> R1 >> L2 >> R2;
        ++ L1, ++ R1;
        For(x, L1, L2) {
            For(y, R1, R2) {
                a[x][y] = 1;
            }
        }
    }
    int ans = 0;
    For(x, 0, 100) {
        For(y, 0, 100) {
            ans += a[x][y];
        }
    }
    cout << ans << endl;
    return 0;
}
```

- 题面

- 题面
- 简化题意：两坨区间的重叠长度

- 题面
- 简化题意：两坨区间的重叠长度
- 数据范围很小，怎么暴力都行

- 题面
- 简化题意：两坨区间的重叠长度
- 数据范围很小，怎么暴力都行
- 下面代码采用先标记 A 的可聊时间，再统计 B 的可聊时间中 A 的可聊时间之和

第十四届. 买菜

```
#include <bits/stdc++.h>
#define For(i, l, r) for (int i=l; i<=r; ++i)
using namespace std;

int n, a, b, c, d, liao[1000010];
signed main() {
    cin >> n;
    For(i, 1, n) {
        cin >> a >> b;
        For(j, a, b-1) {
            liao[j] = 1;
        }
    }
    int ans = 0;
    For(i, 1, n) {
        cin >> c >> d;
        For(j, c, d-1) {
            ans += liao[j];
        }
    }
    cout << ans << endl;
    return 0;
}
```

- 题面

- 题面
- 考虑建两个栈，一个存数，一个存符号

- 题面
- 考虑建两个栈，一个存数，一个存符号
- 考虑运算优先级，碰到优先级高的直接算，优先级低的先压到栈里

第十六届. 二十四点

- 题面
- 考虑建两个栈，一个存数，一个存符号
- 考虑运算优先级，碰到优先级高的直接算，优先级低的先压到栈里
- 最后把栈内元素清空计算答案

第十六届.二十四点

```
#include <bits/stdc++.h>
#define For(i, l, r) for (int i=l; i<=r; ++i)
using namespace std;

int T, num[5];
char c[10], opt[5];
signed main() {
    cin >> T;
    while (T -- > 0) {
        scanf("%s", c+1);
        int n = 7, topn = 0, topc = 0;
        num[++ topn] = c[1] - '0';
        for (int i=2; i<=n; i+=2) {
            if (c[i] == '+' || c[i] == '-') {
                opt[++ topc] = c[i];
                num[++ topn] = c[i+1] - '0';
            }
            else {
                if (c[i] == '/') num[topn] = num[topn] / (c[i+1] - '0');
                else num[topn] = num[topn] * (c[i+1] - '0');
            }
        }
        for (int i=1; i<=topc; ++i) {
            if (opt[i] == '+') num[i+1] = num[i] + num[i+1];
            else num[i+1] = num[i] - num[i+1];
        }
        cout << (num[topn] == 24 ? "Yes" : "No") << endl;
    }
    return 0;
}
```

- 题面

第十八届. 回收站选址

- 题面
- 记录每个坐标是否有垃圾

第十八届. 回收站选址

- 题面
- 记录每个坐标是否有垃圾
- 考虑到坐标值域很大，不能用二维数组记录，所以采用 map

第十八届. 回收站选址

```
#include <bits/stdc++.h>
#define For(i, l, r) for (int i=l; i<=r; ++i)
using namespace std;

const int N = 1005;
const int dx[4] = {0, 0, -1, 1},
          dy[4] = {1, -1, 0, 0};
const int dx2[4] = {1, 1, -1, -1},
          dy2[4] = {1, -1, 1, -1};
int n, x[N], y[N], ans[5];
map<pair<int, int>, int> mp;
signed main() {
    cin >> n;
    For(i, 1, n) {
        cin >> x[i] >> y[i];
        mp[{x[i], y[i]}] = 1;
    }
    For(i, 1, n) {
        int cnt = 0;
        For(k, 0, 3) {
            int nx = x[i] + dx[k],
                ny = y[i] + dy[k];
            cnt += mp[{nx, ny}];
        }
        if (cnt == 4) {
            cnt = 0;
            For(k, 0, 3) {
                int nx = x[i] + dx2[k],
                    ny = y[i] + dy2[k];
```


第十八届. 回收站选址

```
        cnt += mp[{nx, ny}];
    }
    ++ ans[cnt];
}
}
For(i, 0, 4) {
    cout << ans[i] << endl;
}
return 0;
}
```

第二十六届. 寻宝！大冒险！

- 题面

第二十六届. 寻宝! 大冒险!

- 题面
- 注意到 L 很大 n 却很小, 所以这是一张稀疏图

第二十六届. 寻宝！大冒险！

- 题面
- 注意到 L 很大 n 却很小，所以这是一张稀疏图
- 所以我们不需要枚举每一个坐标作为左下角

第二十六届. 寻宝！大冒险！

- 题面
- 注意到 L 很大 n 却很小，所以这是一张稀疏图
- 所以我们不需要枚举每一个坐标作为左下角
- 只需要将藏宝图中最左下角的树和已知的树去匹配，check 剩下藏宝图中的树位置对不对就行

第二十六届. 寻宝！大冒险！

- 题面
- 注意到 L 很大 n 却很小，所以这是一张稀疏图
- 所以我们不需要枚举每一个坐标作为左下角
- 只需要将藏宝图中最左下角的树和已知的树去匹配，check 剩下藏宝图中的树位置对不对就行
- 这样时间复杂度就是 $O(n^2 + S^2)$

第二十六届. 寻宝! 大冒险!

```
#include <bits/stdc++.h>
#define For(i, l, r) for (int i=l; i<=r; ++i)
using namespace std;
int n, L, S, k, a[55][55], x[1005], y[1005];
signed main() {
    cin >> n >> L >> S;
    For(i, 1, n) {
        cin >> x[i] >> y[i];
    }
    for (int i=S; i>=0; --i) {
        for (int j=0; j<=S; ++j) {
            cin >> a[i][j]; k += a[i][j];
        }
    }
    int ans = 0;
    For(i, 1, n) {
        if (x[i] + S > L || y[i] + S > L) continue;
        int cnt = 0;
        bool flag = true;
        For(j, 1, n) {
            if (x[i] <= x[j] && x[j] <= x[i] + S && y[i] <= y[j] && y[j] <= y[i] + S) {
                if (!a[x[j]-x[i]][y[j]-y[i]]) {
                    flag = false; break;
                }
                ++ cnt;
            }
        }
        if (flag && cnt == k) ++ ans;
    }
    cout << ans << endl;
```

- Thanks for Listening.
- Good Luck and Have Fun.