

# 折半搜索

## 人员

杨瑾硕、陈洛冉 到课, 司云心 线上

## 上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/2584>

#	用户名	姓名	编程分	时间	A	B	C
1	yuzijia1	于子珈	200	342	100	100	
2	yangjinshuo	杨瑾硕	200	430	100	100	
3	zhaoxiyu	赵熙羽	200	833	100	100	
4	zhouziyi	周子一	100	406	100		

## 本周作业

<https://cppoj.kids123code.com/contest/2684> (课上讲了 A ~ B 题, 课后作业是 C 题)

## 课堂表现

今天课上讲了折半搜索这个算法, 同学们对  $2^n$  的搜索写的不熟, 课下需要再多复习一下搜索的基础知识。

## 课堂内容

### [CEOI 2015] 世界冰球锦标赛 (Day2)

折半搜索模板题, 直接  $2^{40}$  会超时, 所以对前一部分进行  $2^{20}$  的搜索, 对后一部分也进行  $2^{20}$  的搜索, 然后把两次的结果记录下来最终求答案

```
#include <bits/stdc++.h>
#define int long long

using namespace std;

const int maxn = 2e6 + 5;
int w[maxn];
int v1[maxn], v2[maxn];
int cnt1 = 0, cnt2 = 0;

void dfs(int u, int ed, int sum, int v[], int& cnt) {
    if (u == ed+1) {
        ++cnt; v[cnt] = sum;
        return;
    }
    for (int i = 0; i <= 1; ++i) {
        v[i] = sum;
        dfs(u + 1, ed, sum + w[u], v, cnt);
    }
}
```

```

    }

    dfs(u+1, ed, sum, v, cnt);
    dfs(u+1, ed, sum+w[u], v, cnt);
}

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

    int mid = n / 2;
    dfs(1, mid, 0, v1, cnt1); dfs(mid+1, n, 0, v2, cnt2);
    sort(v1+1, v1+cnt1+1); sort(v2+1, v2+cnt2+1);

    int res = 0;
    for (int i = 1; i <= cnt1; ++i) {
        int pos = upper_bound(v2+1, v2+cnt2+1, m-v1[i]) - v2 - 1;
        res += pos;
    }
    cout << res << endl;
    return 0;
}

```

## XOR on Grid Path

从左上角往右下做一次搜索, 再从右下角往左上做一次搜索, 把搜到的结果用 map 存下来, 最后统计方案数即可

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 20 + 5;
int w[maxn][maxn];
map<int, int> mp1[maxn][maxn], mp2[maxn][maxn];
int n;

void dfs1(int x, int y, int value) {
    if (x+y == n+1) { mp1[x][y][value^w[x][y]]++; return; }

    dfs1(x+1, y, value^w[x][y]);
    dfs1(x, y+1, value^w[x][y]);
}

void dfs2(int x, int y, int value) {
    if (x+y == n+2) { mp2[x][y][value^w[x][y]]++; return; }

    dfs2(x-1, y, value^w[x][y]);
    dfs2(x, y-1, value^w[x][y]);
}

```

```
int main()
{
    cin >> n;
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= n; ++j) cin >> w[i][j];
    }

    dfs1(1, 1, 0); dfs2(n, n, 0);

    LL res = 0;
    for (int x = 1, y = n; x <= n; ++x, --y) {
        for (auto it : mp1[x][y]) {
            int val = it.first, cnt = it.second;
            res += (LL)cnt * mp2[x+1][y][val];
            res += (LL)cnt * mp2[x][y+1][val];
        }
    }
    cout << res << endl;
    return 0;
}
```