

# C4-J 题题解

王家豪

November 8, 2022

## J 题题意

给定  $n \times m$  的矩阵  $a_{i,j}$ , 要从  $(1, 1)$  走到  $(n, m)$ , 每次可以向下或向右移动, 将沿途经过的数进行按位与运算, 得到最终结果为  $A$ , 求  $A$  的最大值。

如果矩阵中的数都为 0 或 1，那么结果是不是容易计算？

如果希望最终得数  $A$  尽可能的大，是不是应该让  $A$  二进制表示的高位尽可能为 1？

如果某一个高位已经计算过，如何在此基础上继续考虑低位？

# 核心代码

```
int vis = 0;
int ans = 0;
for (int d = 29; d >= 0; d--) {
    vis += 1 << d;
    dp[1][1] = (a[1][1] & vis) == vis;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            if (i == 1 && j == 1) continue;
            if ((a[i][j] & vis) == vis &&
                (dp[i][j - 1] == 1 || dp[i - 1][j] == 1)) {
                dp[i][j] = 1;
            } else {
                dp[i][j] = 0;
            }
        }
    }
    if (dp[n][m]) {
        ans += 1 << d;
    } else {
        vis -= 1 << d;
    }
}
```

时间复杂度:  $O(tnm \log A)$

# Opt 完整代码

```
#include <stdio.h>

#define N 205

int a[N][N];
int dp[N][N];

int main() {
    int tt;
    scanf("%d", &tt);
    while (tt--) {
        int n, m;
        scanf("%d%d", &n, &m);
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= m; j++) {
                scanf("%d", &a[i][j]);
            }
        }
        int vis = 0;
        int ans = 0;
        for (int d = 29; d >= 0; d--) {
            vis += 1 << d;
            dp[1][1] = (a[1][1] & vis) == vis;
            for (int i = 1; i <= n; i++) {
                for (int j = 1; j <= m; j++) {
                    if (i == 1 && j == 1) continue;
                    if ((a[i][j] & vis) == vis && (dp[i][j - 1] == 1 || dp[i - 1][j] == 1)) {
                        dp[i][j] = 1;
                    } else {
                        dp[i][j] = 0;
                    }
                }
            }
        }
        if (dp[n][m]) {
```

```
        ans += 1 << d;
    } else {
        vis -= 1 << d;
    }
}
printf("%d\n", ans);
}
return 0;
}
```