Gold Mine Problem

从后往前推算

```
#include <iostream>
#include <vector>
#include <stdio.h>
#include <climits>

using namespace std;

// dp[i][j] 表示M[i][j]位置的最大黄金数量
// 从最后一列，第一行开始，往前推算到第一列，所有行，计算出 dp[row][0]的值
int getMaxGold(vector<vector<int>> gold, int n, int m) {
    vector<vector<int>> dp(n, vector<int>(m));
    for (int col = m - 1; col >= 0; col--) {
        for (int row = 0; row < n; ++row) {
            int rightGold = col == m - 1 ? 0 : dp[row][col + 1];
            int rightUpGold = (row == 0 || col == m - 1) ? 0 : dp[row - 1][col + 1];
            int rightDownGold = (row == n - 1 || col == m - 1) ? 0 : dp[row + 1][col + 1];
            dp[row][col] = gold[row][col] + max(rightGold, max(rightUpGold, rightDownGold));
        }
    }
    int maxGold = dp[0][0];
    for (int i = 1; i < n; ++i)
        maxGold = max(maxGold, dp[i][0]);
    return maxGold;

}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int n, m, val;
        scanf("%d %d", &n, &m);
        vector<vector<int>> gold(n, vector<int>(m));
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                scanf("%d", &val);
                gold[i][j] = val;
            }
        }
        int res = getMaxGold(gold, n, m);
        printf("%d\n", res);

    }
    return 0;
}
```