

## 174. Dungeon Game

题目描述: <https://leetcode.com/problems/dungeon-game/>

给定m\*n的矩阵代表地牢，存储着打入每个地牢所需的hp值。

正的值代表 加血

负的值代表 减血 0代表，不加不减。

已知公主在右下角，骑士从左上角开始救人，请问要救到公主至少需要的hp值

解题思路：

倒着dp。

f[i][j]代表从该位置到救到公主所需的hp值。 则：

$f[i][j] = \min(f[i+1][j], f[i][j+1]) - \text{matrix}[i][j]$

如果f[i][j] >= 0，则证明无需自带hp靠加血就行，因此f[i][j] = 1；

代码：

```
class Solution {
public:
    int calculateMinimumHP(vector<vector<int>>& dungeon) {
        int m = dungeon.size();
        if(m == 0) return 0;
        int n = dungeon[0].size();
        vector<vector<int>> f(m+1, vector<int>(n+1, INT_MAX));
        f[m][n-1] = f[m-1][n] = 1;
        for(int i = m-1; i >= 0; i--) {
            for(int j = n-1; j >= 0; j--) {
                int needHP = min(f[i+1][j], f[i][j+1]) - dungeon[i][j];
                f[i][j] = needHP <= 0 ? 1 : needHP;
            }
        }
        return f[0][0];
    }
};
```

代码2：

```
class Solution {
public:
    int calculateMinimumHP(vector<vector<int>>& dungeon) {
        int m = dungeon.size();
        if(m == 0) return 0;
        int n = dungeon[0].size();
        vector<int> f(n+1, INT_MAX);
        f[n-1] = 1;
        for(int i = m-1; i >= 0; i--) {
            for(int j = n-1; j >= 0; j--) {
                int needHP = min(f[j], f[j+1]) - dungeon[i][j];
                f[n] = INT_MAX;
                f[j] = needHP <= 0 ? 1 : needHP;
            }
        }
        return f[0];
    }
};
```