

## 第29题——机器人的运动范围

地上有一个m行n列的方格，从坐标 [0,0] 到坐标 [m-1,n-1]。一个机器人从坐标 [0, 0] 的格子开始移动，它每次可以向左、右、上、下移动一格（不能移动到方格外），也不能进入行坐标和列坐标的数位之和大于k的格子。例如，当k为18时，机器人能够进入方格 [35, 37]，因为  $3+5+3+7=18$ 。但它不能进入方格 [35, 38]，因为  $3+5+3+8=19$ 。请问该机器人能够到达多少个格子？

示例 1:

输入:  $m = 2, n = 3, k = 1$   
输出: 3

示例 2:

输入:  $m = 3, n = 1, k = 0$   
输出: 1

提示:

- $1 \leq n, m \leq 100$
- $0 \leq k \leq 20$

## DFS(1)

首先必须维护一个二维数组,初始化为0 调用递归 递归的内容: 判断条件(1->是否越界, 2->是否已访问, 3->位数和是否大于k) 是->返回; 否->count++,递归右和下 因为我们从左上角(0,0)出发,所以只能向右和向下走 因此,只需要递归i+1和j+1即可 返回count;

```
class Solution {
public:
    int count=0;
    int movingCount(int m, int n, int k) {
        if(k==0)
            return 1;
        vector<vector<int>> board(m, vector<int>(n,0)); //维护一个二维数组
        //调用递归
        dfs(board,0,0,m,n,k);
        return count;
    }
    int bit(int m,int n)
    {
        return (m/10+n/10)+(m%10+n%10);
    }
    void dfs(vector<vector<int>>&board,int i,int j,int m,int n,int k)
    {
        if(i>=m || j>=n || i<0 || j<0 || bit(i,j)>k || board[i][j]==1)
        {
```

```
        return;//不符合条件则返回即可
    }
    count++;
    board[i][j]=1;
    dfs(board,i+1,j,m,n,k);//向下
    dfs(board,i,j+1,m,n,k);//向右
}
};
```

作者: sakura7301

链接: <https://leetcode-cn.com/problems/ji-qi-ren-de-yun-dong-fan-wei-lcof/solution/bi-jiao-jian-ji-de-dfssuan-fa-by-sakura7-c97j/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。