

机器人的运动范围

地上有一个m行n列的方格，一个机器人从坐标(0,0)的格子开始移动，它每一次可以向左、右、上、下移动一格，但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)中，因为 $3 + 5 + 3 + 8 = 19$ 。请问该机器人能够到达多少个格子。

主要思路：主要是用回溯法。机器人从(0,0)开始移动，当机器人准备移动到坐标(x,y)时，首先判断该坐标是否满足条件，若满足条件则进入该格子，再判断是否能进入周围相邻的格子。

```
public class B13 {
    //约束值、方格行数、方格列数
    public int movingCount(int threshold, int rows, int cols) {
        boolean [] visited = new boolean[rows * cols];
        for(int i = 0; i < visited.length; i++) {
            visited[i] = false;
        }
        int count = movingCountCore(threshold, rows, cols, 0, 0, visited);
        return count;
    }

    public int movingCountCore(int threshold, int rows, int cols, int row, int col, boolean [] visited) {
        int count = 0;
        if(check(threshold, rows, cols, row, col, visited)) {
            visited[row * cols + col] = true;

            count = 1 + movingCountCore(threshold, rows, cols, row - 1, col, visited) +
                    movingCountCore(threshold, rows, cols, row, col - 1, visited) +
                    movingCountCore(threshold, rows, cols, row + 1, col, visited) +
                    movingCountCore(threshold, rows, cols, row, col + 1, visited);
        }
        return count;
    }
    //判断这个点是不是可以走
    boolean check(int threshold, int rows, int cols, int row, int col, boolean [] visited) {
        if(row >= 0 && row <= rows && col >= 0 && col < cols &&
            getDigitSum(row) + getDigitSum(col) <= threshold
            && !visited[row * cols + col]) {
            return true;
        }

        return false;
    }
    //返回一个数值的位数之和
    public int getDigitSum(int number) {
        int sum = 0;
        while (number > 0) {
            sum += number % 10;
            number /= 10;
        }
        return sum;
    }
}
```