使用深度优先搜索算法（dfs）时，前提是

每一步的操作要很相似，用的是递归实现（我是这么实现的），这样的话要进行每一步要进行不同的操作要用if else判断，每一步相似度不高的话就要写很多if else会很复杂，所以当前这一步要与下一步尽可能相似。然后dfs有点暴力，它要搜索所有可能的解，根据实际应用我们可以对它相应地进行减枝，比如找到迷宫出口，就不进行搜索了。

写dfs函数首先要确定结束条件，然后当前要做什么，确定下一步要做什么，递归，递归完要做什么，所以

Void dfs（）

{

if（结束）

{

Return

}

for（）

{

.....

dfs();

。。。

}

}

dfs（）；