实验报告

【实验名称】 实验二 八数码问题求解

【实验目的】

1. 学习广度优先搜索算法；
2. 学习深度优先搜索算法；
3. 学习启发式搜索算法；

【实验原理】

1.问题的提出

穷举搜索法,从理论上讲,似乎可以解决任何状态空间的搜索问题,但实践表明,穷举搜索只能解决一些状态空间很小的简单问题,而对于那些大状态空间问题,穷举搜索就不能胜任了。因为大空间问题往往会导致“组合爆炸”。

上述困难迫使人们不得不寻找更有效的搜索方法,即提出了启发式搜索策略。

2.启发性信息

启发式搜索就是利用启发性信息进行制导的搜索。启发性信息就是有利于尽快找到问题之解的信息。按其用途划分,启发性信息一般可分为以下三类:

(1)用于扩展节点的选择,即用于决定应先扩展哪一个节点,以免盲目扩展。

(2)用于生成节点的选择,即用于决定应生成哪些后续节点,以免盲目地生成过多无用节点。

(3)用于删除节点的选择,即用于决定应删除哪些无用节点,以免造成进一步的时空浪费。

3.启发函数

在启发式搜索中,通常用所谓启发函数来表示启发性信息。启发函数是用来估计搜索树上节点x与目标节点S接近程度的一种函数,通常记为h(x)。

如何定义一个启发函数呢?启发函数并无固定的模式,需要具体问题具体分析。通常可以参考的思路有:一个节点到目标节点的某种距离或差异的度量;一个节点处在最佳路径上的概率;或者根据经验的主观打分,等等。例如,对于八数码难题,用h(x)就可表示节点x的数码格局同目标节点相比数码不同的位置个数。

启发函数设定。对于八数码问题，可以利用棋局差距作为一个度量。搜索过程中，差距会逐渐减少，最终为零，为零即搜索完成，得到目标棋局。

4.启发式搜索算法

启发式搜索要用启发函数来导航,其搜索算法就要在状态图一般搜索算法基础上再增加启发函数值的计算与传播过程,并且由启发函数值来确定节点的扩展顺序。

5.输入数据

输入初始棋盘和目标棋盘。随意输入，但是必须是0~8组成的棋盘且同一数字不得出现两次。

6.输出数据

1. 输出每一步的棋盘，并计算其估计函数f(x)以及启发函数h(x)、深度d(x)；
2. 如果没有解，则提示没有路径。

7.算法流程

搜索失败，退出

N能否扩展（朝上下左右方向可以移动）？

对于子节点，OPEN表、CLOSED表中有N或N的父辈节点？

N=目标棋盘？

OPEN表为空或超出扩展次数上限？

计算剩余子节点的f(x)值，将这些节点配上指向N的指针插入OPEN表中，然后按f(x)值对OPEN表中的节点排序。

删除这些节点

生成子节点

搜索成功，退出

从OPEN表中取出第一个棋盘N放入COLSED表中

把起始棋盘放OPEN表中

输入起始棋盘和目标棋盘

Y

N

Y

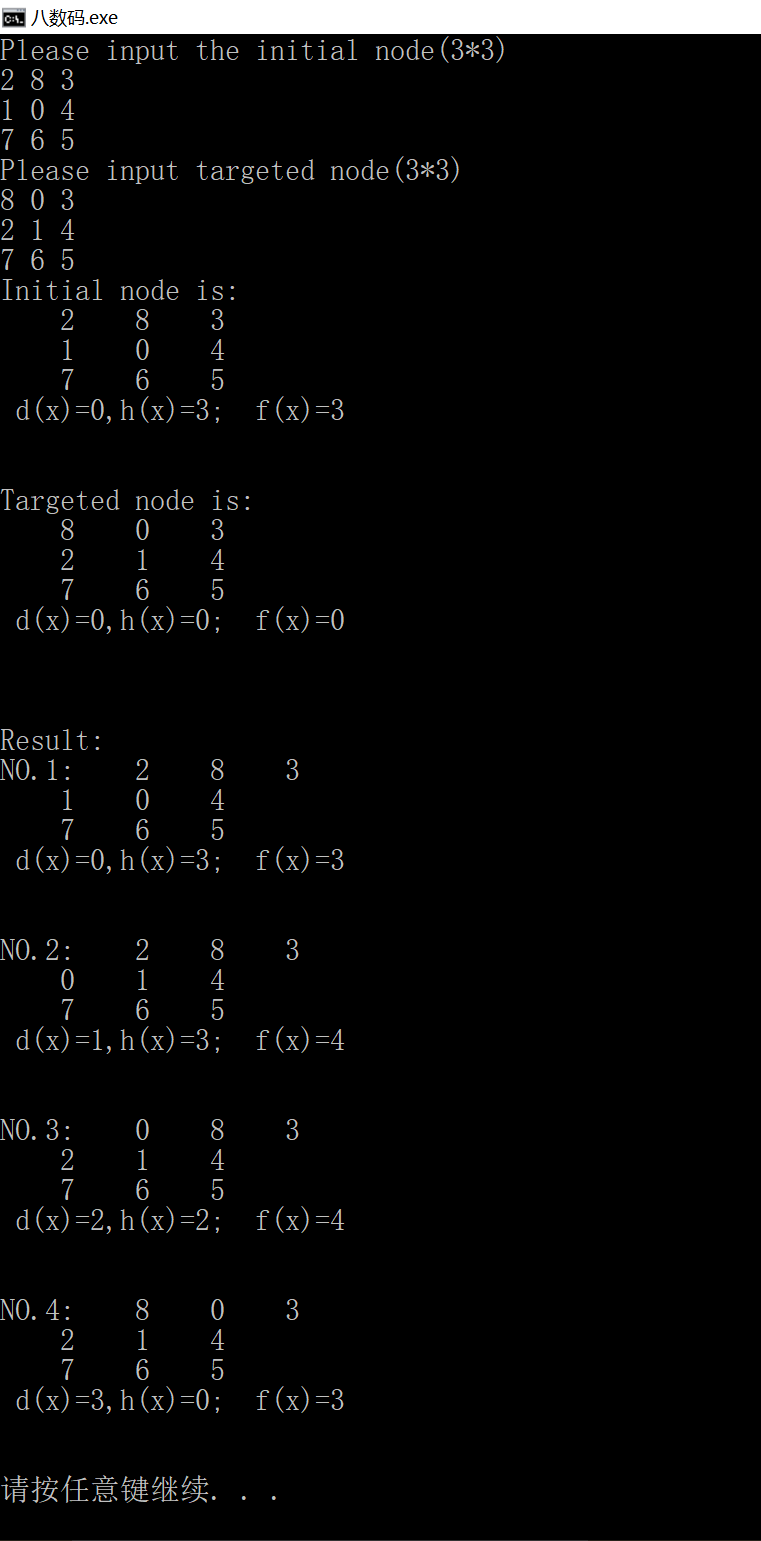
N

N

Y

Y

N

【实验结果】