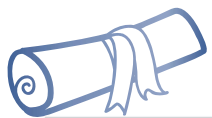




A算法求解八数码

人工智能基础——实验一





八数码问题

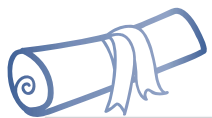
- 状态：描述8个棋子和空位在棋盘的9个方格上的分布情况。其中，任何状态都可以被指定为初始状态。
- 操作符：产生4个行动，即上下左右移动
- 目标测试：用来检测状态是否能匹配上给定的目标状态。
- 路径费用函数：每一步的费用为1，因此整个路径的费用是路径中的步数。
- 问题描述：给定任意一个初始状态，要求找到一种搜索策略，用尽可能少的步数得到上图的目标状态。

2	8	3
1		4
7	6	5

(a)初始状态

1	2	3
8		4
7	6	5

(b)目标状态



A算法

A算法特点在于对估价函数 f 的定义上。对于一般的启发式图搜索，总是选择估价函数 f 值最小的节点作为扩展节点。

估价函数：

$$f(n) = g(n) + h(n)$$

- $g(n)$ 为初始状态到状态 n 是已付出的实际代价；
- $h(n)$ 是从状态 n 到目标状态的最优路径的估计代价，而搜索的启发式信息主要由 $h(n)$ 决定。

$h(n)$ 比重大：降低搜索工作量，但可能导致找不到最优解；

$h(n)$ 比重小：一般导致工作量加大，极限情况下变为盲目搜索，但可能可以找到最优解。

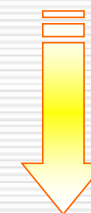
5.4.2 启发信息和估价函数

■ 例5.7 八数码问题的启发函数:

- 启发函数1: 取一棋局与目标棋局相比, 其位置不符的**数码数目**, 例如 $h(S_0) = 5$;
- 启发函数2: 各数码移到目标位置所需移动的**距离的总和**, 例如 $h(S_0) = 6$;
- 启发函数3: 对每一对逆转数码乘以一个倍数, 例如3倍, 则 $h(S_0) = 3$;
- 启发函数4: 将位置不符数码数目的总和与3倍数码逆转数目相加, 例如 $h(S_0) = 8$ 。

2	1	3
7	6	4
	8	5

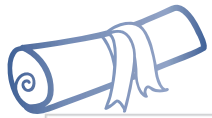
初始棋局



1	2	3
8		4
7	6	5

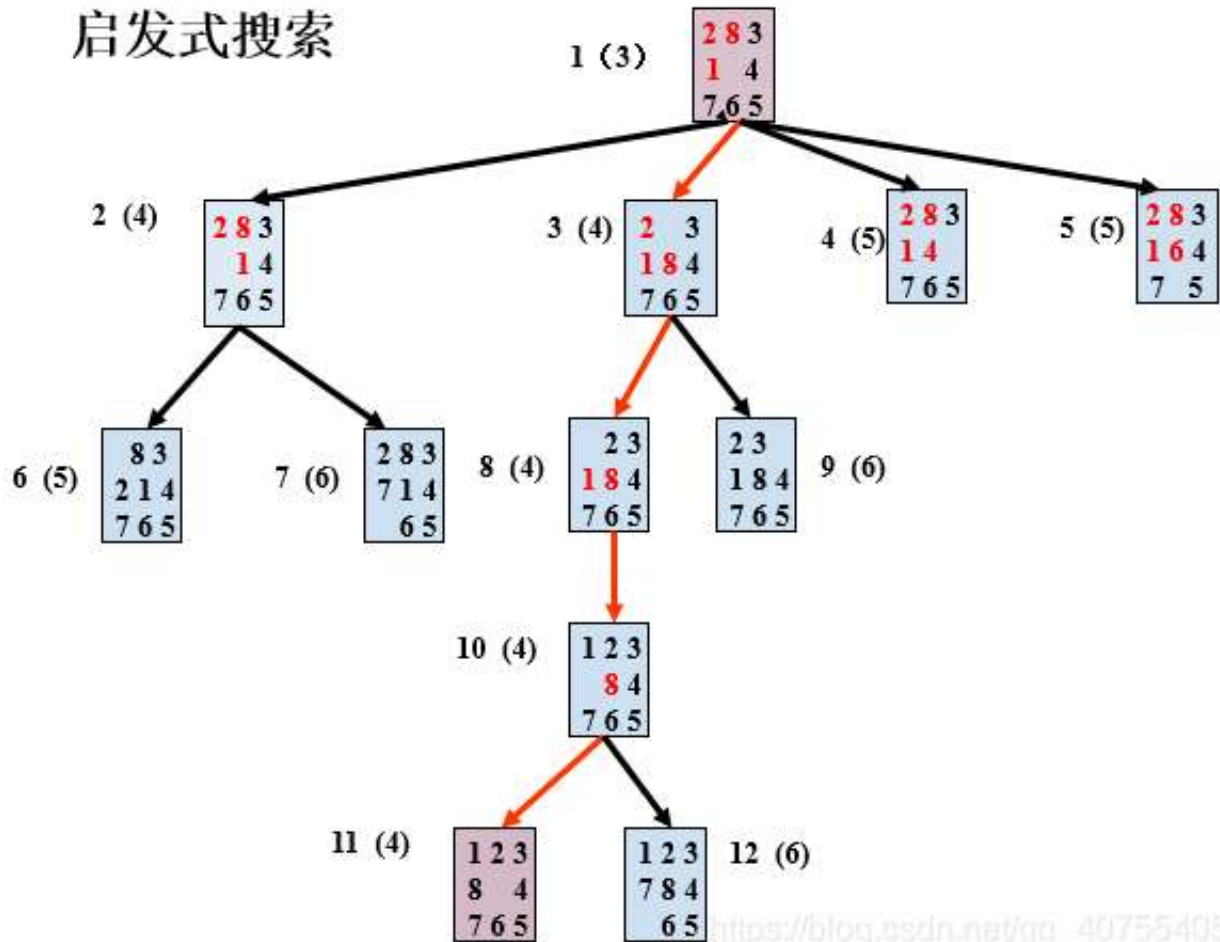
目标棋局

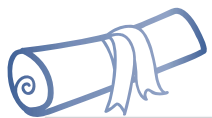
Char 5. pp.4



A*算法

启发式搜索





A算法

- OPEN表保存所有已生成而未考察的节点
- CLOSED表中记录已访问过的节点。

1. 将起始点加入open表
2. 当open表不为空时:
3. 寻找open表中f值最小的点current
4. 如果current是终止点，则找到结果，程序结束。
5. 否则，open表移出current，对current表中的每一个临近点:
6. 若它不可走或在close表中，略过
7. 若它不在open表中，加入。
8. 若它在open表中，计算g值，若g值更小，替换其父节点为current，更新它的g值。
9. . 若open表为空，则路径不存在。



实验要求

1. 实现A算法;
2. 统计达到目标状态是走的路径长度，并可按要求展示中间结果
3. 按要求书写实验报告。

2	8	3
1		4
7	6	5

(a)初始状态

1	2	3
8		4
7	6	5

(b)目标状态