



程序设计实习

郭炜 微博 <http://weibo.com/guoweiofpku>

<http://blog.sina.com.cn/u/3266490431>

刘家瑛 微博 <http://weibo.com/pkuliujiaying>



北京大学
PEKING UNIVERSITY

信息科学技术学院《程序设计实习》 郭炜 刘家瑛

广度优先搜索

八数码问题

八数码 (POJ1077)

□ 八数码问题是人工智能中的经典问题

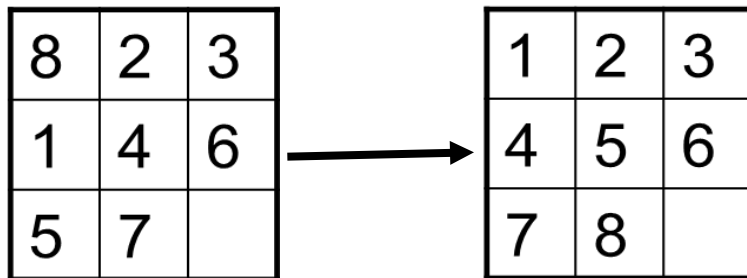
有一个3*3的棋盘，其中有0-8共9个数字，0表示空格，其他的数字可以和0交换位置。求由初始状态到达目标状态

1 2 3

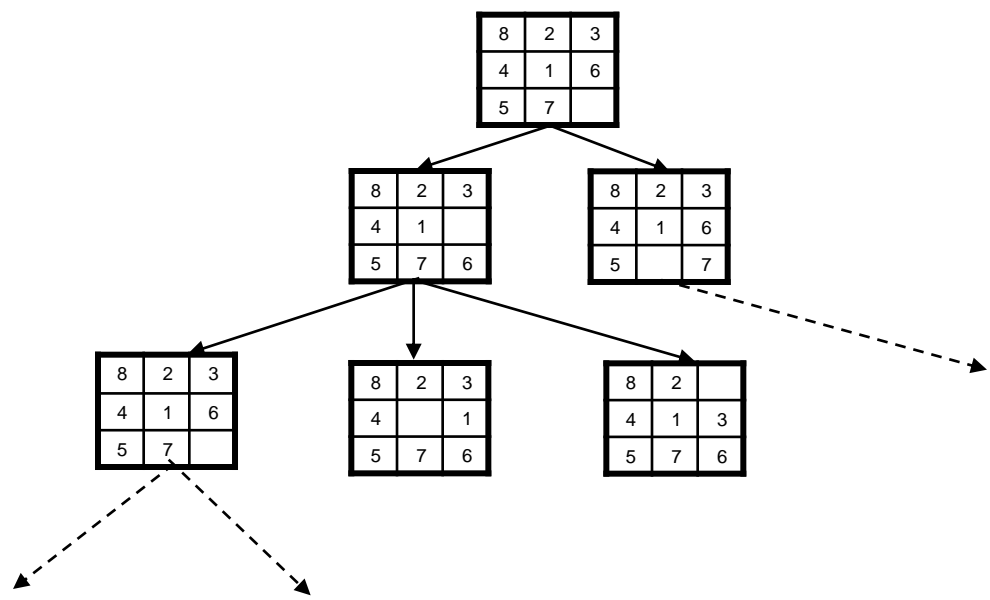
4 5 6

7 8 0

的步数最少的解。

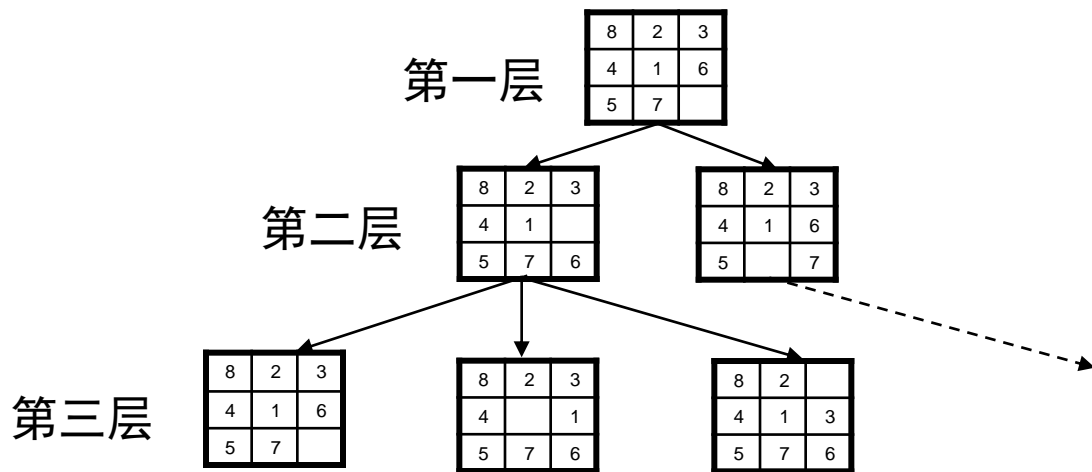


● 状态空间



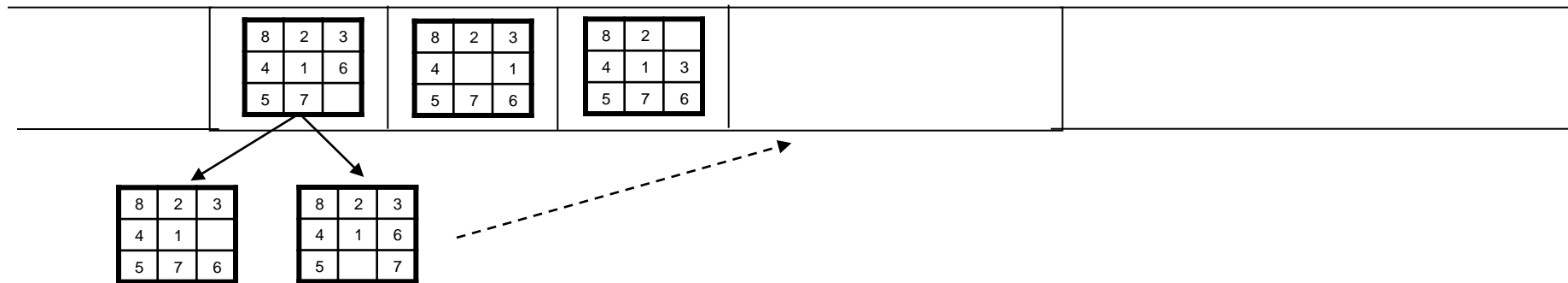
● 广度优先搜索 (bfs)

- 优先扩展浅层节点(状态)，逐渐深入



● 广度优先搜索

- 用**队列**保存待扩展的节点
- 从队首取出节点，扩展出的新节点放入队尾，直到队首出现目标节点（问题的解）



● 广度优先搜索的代码框架

```
BFS()
```

```
{
```

```
    初始化队列
```

```
    while(队列不为空且未找到目标节点)
```

```
    {
```

```
        取队首节点扩展，并将扩展出的非重复节点放入队尾；
```

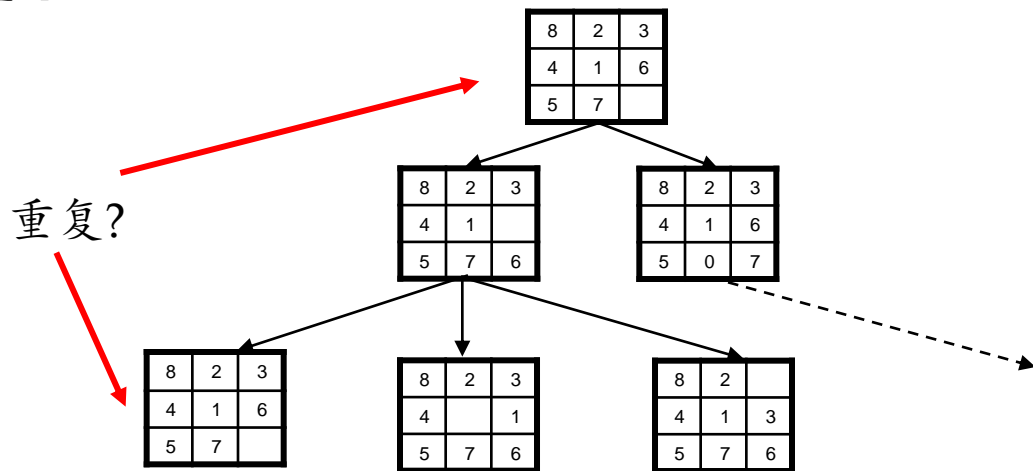
```
        必要时要记住每个节点的父节点；
```

```
    }
```

```
}
```

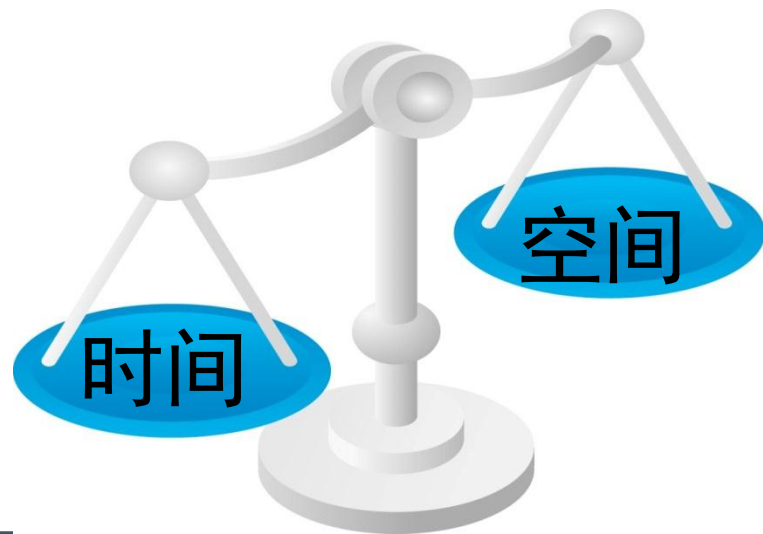
关键问题：判重

- 新扩展出的节点如果和以前扩展出的节点相同，则这个新节点就不必再考虑
- 如何判重？



关键问题：判重

- 状态(节点) 数目巨大，如何存储？
- 怎样才能较快判断一个状态是否重复？



用合理的编码表示“状态”，减小存储代价

- 方案一：

8	2	3
4	1	6
5	7	

每个状态用一个字符串存储,
要9个字节, 太浪费了!!!

用合理的编码表示“状态”，减小存储代价

● 方案二：

8	2	3
4	1	6
5	7	

- 每个状态对应于一个9位数，则该9位数最大为876,543,210，小于 2^{31} ，则int 就能表示一个状态。
- 判重需要一个标志位序列，每个状态对应于标志位序列中的1位，标志位为0表示该状态尚未扩展，为1则说明已经扩展过了
- 标志位序列可以用字符数组a存放。a的每个元素存放8个状态的标志位。最多需要876,543,210位，因此a数组需要 $876,543,210 / 8 + 1$ 个元素，即 109,567,902 字节
- 如果某个状态对应于数x，则其标志位就是a[x/8]的第x%8位
- 空间要求还是太大!!!!

用合理的编码表示“状态”，减小存储代价

- 方案三：

8	2	3
4	1	6
5	7	

- 将每个状态的字符串形式看作一个9位九进制数，则该9位数最大为 $876543210_{(9)}$ ，即 $381367044_{(10)}$ 需要的标志位数目也降为 $381367044_{(10)}$ 比特，即47,670,881字节。
- 如果某个状态对应于数 x ，则其标志位就是 $a[x/8]$ 的第 $x\%8$ 位
- 空间要求还是有点大！！！！

用合理的编码表示“状态”，减小存储代价

- 方案三：

8	2	3
4	1	6
5	7	

- 状态数目一共只有 $9!$ 个，即 $362880_{(10)}$ 个，怎么会需要 $876543210_{(9)}$ 即 $381367044_{(10)}$ 个标志位呢？

用合理的编码表示“状态”，减小存储代价

- 方案三：

8	2	3
4	1	6
5	7	

- 状态数目一共只有 $9!$ 个，即 $362880_{(10)}$ 个，怎么会需要 $876543210_{(9)}$ 即 $381367044_{(10)}$ 个标志位呢？
- 如果某个状态对应于数 x ，则其标志位就是 $a[x/8]$ 的第 $x\%8$ 位
- 因为有浪费！例如， $666666666_{(9)}$ 根本不对应于任何状态，也为其准备了标志位！

用合理的编码表示“状态”，减小存储代价

● 方案四：

8	2	3
4	1	6
5	7	

- 把每个状态都看做'0'-'8'的一个排列，以此排列在全部排列中的位置作为其序号。状态用其排列序号来表示
- 012345678是第0个排列，876543210是第 $9!-1$ 个
- 状态总数即排列总数： $9!=362880$
- 判重用的标志数组a只需要362,880比特即可。
- 如果某个状态的序号是x,则其标志位就是 $a[x/8]$ 的第 $x\%8$ 位

用合理的编码表示“状态”，减小存储代价

- 方案四：

8	2	3
4	1	6
5	7	

- 在进行状态间转移，即一个状态通过某个移动变化到另一个状态时，需要先把int形式的状态（排列序号），转变成字符串形式的状态，然后在字符串形式的状态上进行移动，得到字符串形式的新状态，再把新状态转换成int形式（排列序号）。

用合理的编码表示“状态”，减小存储代价

- 方案四：

8	2	3
4	1	6
5	7	

- 需要编写给定排列（字符串形式）求序号的函数
- 需要编写给定序号，求该序号的排列（字符串形式）的函数

给定排列求序号：

整数 $1, 2 \cdots k$ 的一个排列：

$a_1 \ a_2 \ a_3 \ \cdots a_k$

求其序号

基本思想：算出有多少个排列比给定排列小。

先算1到 a_1-1 放在第1位，会有多少个排列： $(a_1-1) * ((k-1)!)$

再算 a_1 不变，1到 a_2-1 放在第2位(左边出现过的不能再用)，会有多少个排列： $(a_2-1) * ((k-2)!)$

再算 a_1, a_2 不变，1到 a_3-1 放在第3位，会有多少个排列

... 全加起来。 时间复杂度： $O(n^2)$

3241

1, 2放在第一位，有 $2*3! = 12$ 种

3在第一位，1放在第2位，有 $2! = 2$ 种

32? 1放在第3位，有 1种

=>前面共 $12+2+1 = 15$ 种。所以 3241是第16个排列

给定序号n求排列:

1234的排列的第9号

第一位假定是1, 共有 $3!$ 种, 没有到达9, 所以第一位至少是2

第一位是2, 一共能数到 $3!+3!$ 号, ≥ 9 , 所以第一位是2

第二位是1, 21??, 一共能数到 $3!+2! = 8$ 不到9, 所以第二位至少是 3

第二位是3, 23??, 一共能数到 $3!+2!+2! \geq 9$, 因此第二位是3

第三位是1, 一共能数到 $3!+2!+1 = 9$, 所以第三位是1, 第四位是 4

答案: 2314

时间复杂度: $O(n^2)$

- 时间与空间的权衡

- 对于状态数较小的问题，可以用最直接的方式编码以空间换时间
- 对于状态数太大的问题，需要利用好的编码方法以时间换空间
- 具体问题具体分析

用广搜解决八数码问题 (POJ1077)

输入数据:

2 3 4 1 5 0 7 6 8

输出结果:

ulddrurdllurdruldr

输出数据是一个移动序列, 使得移动后
结果变成

1 2 3

4 5 6

7 8

输入样例:

2 3 4

1 5

7 6 8

移动序列中

u 表示使空格上移

d 表示使空格下移

r 表示使空格右移

l 表示使空格左移

八数码问题有解性的判定

- 八数码问题的一个状态实际上是0~8的一个排列，对于任意给定的初始状态和目标，不一定有解，即从初始状态不一定能到达目标状态。
 - 因为排列有奇排列和偶排列两类，从奇排列不能转化成偶排列或相反。
- 如果一个数字0~8的随机排列，用 $F(X)$ ($X \neq 0$) 表示数字X前面比它小的数的个数，全部数字的 $F(X)$ 之和为 $Y = \sum (F(X))$ ，如果Y为奇数则称该排列是奇排列，如果Y为偶数则称该排列是偶排列。
 - 871526340排列的 $Y=0+0+0+1+1+3+2+3=10$ ，10是偶数，所以是偶排列。
 - 871625340排列的 $Y=0+0+0+1+1+2+2+3=9$ 9是奇数，所以是奇排列。
 - 因此，可以在运行程序前检查初始状态和目标状态的奇偶性是否相同，相同则问题可解，应当能搜索到路径。否则无解。

八数码问题有解性的判定

证明：移动0的位置，不改变排列的奇偶性

a1 a2 a3 a4 0 a5 a6 a7 a8 a9

0向上移动：

a1 0 a3 a4 a2 a5 a6 a7 a8 a9