**使用Qlearning实现汉诺塔求解**

主要思路：将汉诺塔中的各个状态进行编码，然后计算出各个状态之间相互转移时的R矩阵和Q矩阵，然后给定任一状态，根据Q矩阵即可找到通向目标的路径。

首先需要定义汉诺塔的各个状态，这里使用如'AAA'来表示状态，三个字母分别表示最小盘，中盘和最大盘在A,B,C的哪个塔中，'AAA'表示三个盘都在A塔中，并且呈小，中，大盘依次向下摆放，共选取状态如下：

states={

1:'AAA',

2:'BAA',

3:'CAA',

4:'BCA',

5:'CBA',

6:'CCA',

7:'BBA',

8:'CCB',

9:'BBC',

10:'ACB',

11:'BCB',

12:'CBC',

13:'CAC',

14:'ABC',

15:'CBC',

16:'ACC',

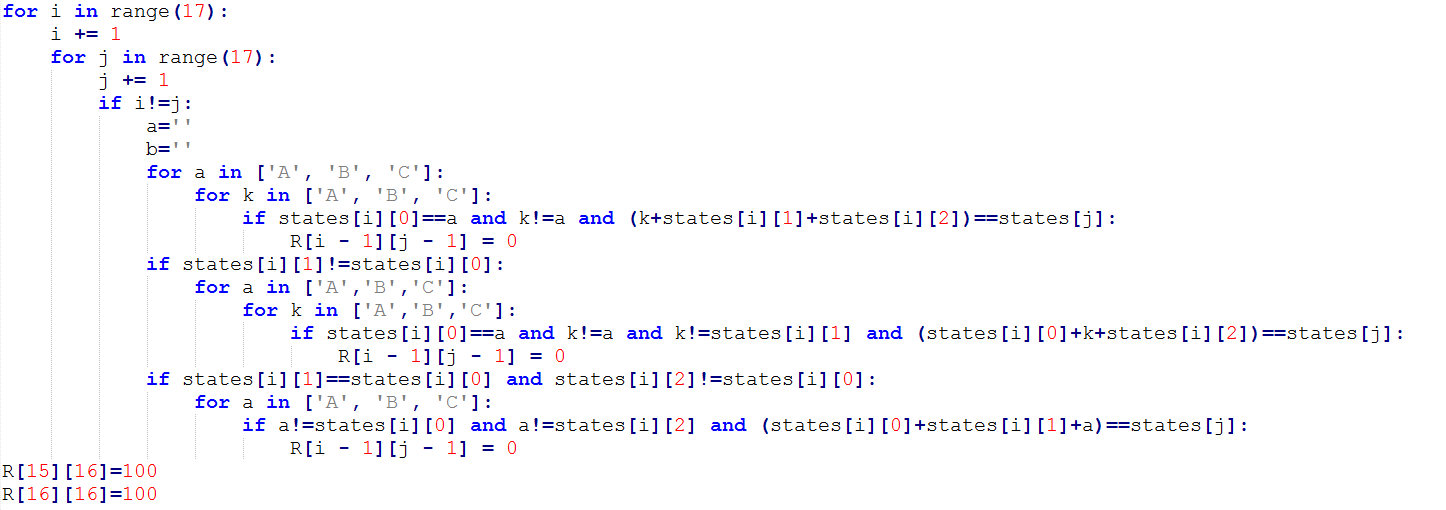
17:'CCC'

}

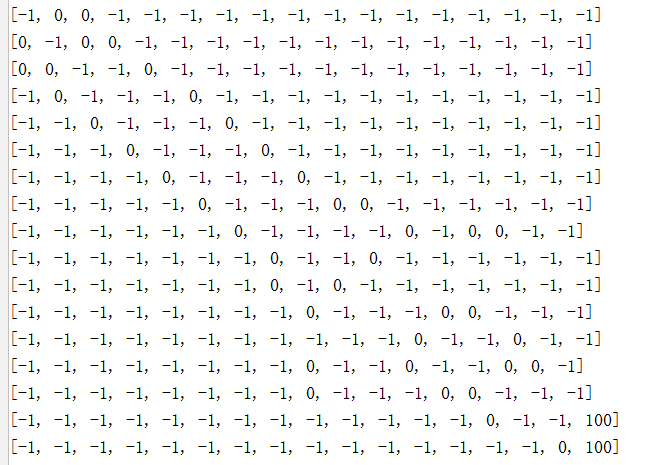
然后需要确定R矩阵和Q矩阵。R矩阵代表奖励矩阵，状态之间能够转移，则赋为0，如果不能转移则赋值为-1，能到达目标状态的状态转移赋值为100。

这里将各个状态之间的转移默认设置为-1，然后找出那些有效的转移赋值为0，最后将能到达目标状态的状态转移赋值为100。在找有效的转移时，有两个汉诺塔状态。分别对三个盘子进行判断，看每个盘子进行转移后，第一个状态是否等于第二个状态，若等于则转移有效。小盘子能够转移的条件：无论何时，小盘子在最上面，能够转移，需要转移到另外的两个塔上。中盘子转移条件：小盘子没有在中盘子上面，中盘子转移到当前没有小盘子也没有中盘子的塔上。大盘子转移条件：中盘子和小盘子在一个塔上，大盘子在另一个塔上，大盘子从当前塔转移到另一个空塔上。

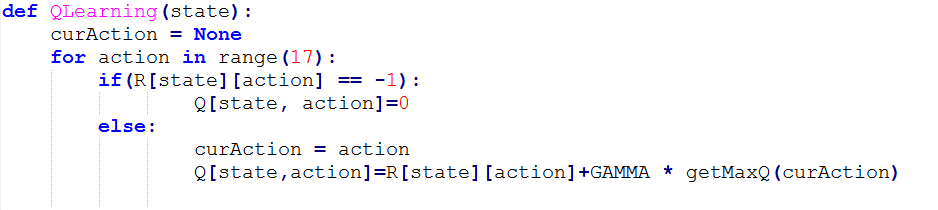
确定R矩阵的代码如下：



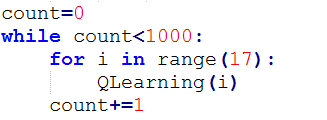
R矩阵的结果如下：



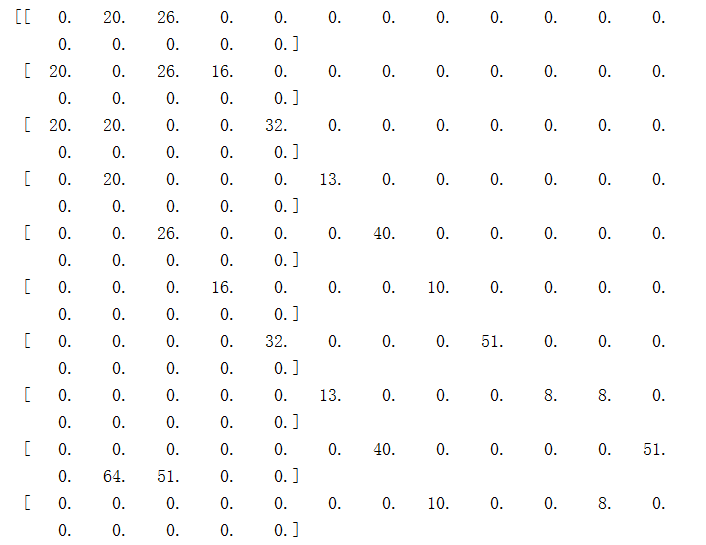
然后需要根据R矩阵确定Q矩阵。只需要对各个状态的各个行动，根据Q矩阵的公式进行更新就可以了。给定某一状态更新其各行动的Q矩阵的值的函数如下：

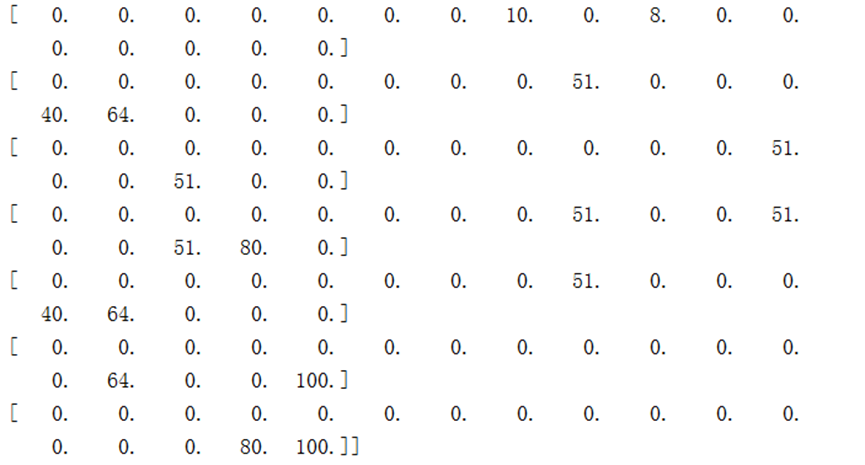


然后可以执行大量的周期，每个周期中，为每个状态调用QLearning函数即可。

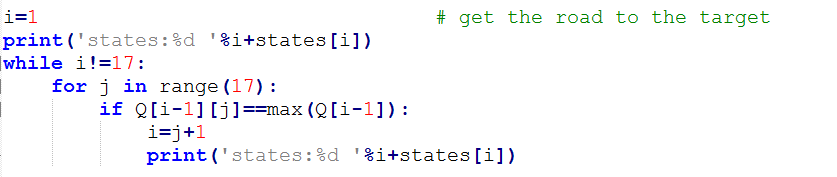


Q矩阵的结果如下：





有了Q矩阵，给定任意状态，可以根据Q矩阵来找到一条通向目标的路径。每次从Q矩阵中找到当前状态下，值最大的行动，然后根据行动更新状态即可。打印从状态1到状态17的路径代码如下。



结果如下：

