四皇后问题

四皇后问题有解，而且有两种解，如下图所示（红色代表皇后位置）。

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

为什么有解？当皇后数小于四时，第一个皇后无论如何放置，剩余皇后放置时都会出现皇后在同一行同一列或者同一斜线的情况，所以当皇后数小于四时问题无解。当皇后数为四时，第一个皇后放在第一列第二行或者第三行，除去其所在的行列及斜线的格子，仍然有空格子，而且某些空格子可以放置不相互冲突的皇后，所以四皇后问题有解。

***Properties of BFS***

Completeness: Yes (if b is finite)

因为BFS会搜索所有节点，所以当目标节点处于某个有限的深度，BFS将找到解决方案

Optimality: Yes only if the path cost is a non-decreasing function of

the depth of the node; not optimal in general

如果路径成本是节点深度的非递减函数，BFS将找到解决方案，但通常不是最优的，因为当节点到根节点的费用和节点的深度成正比时，BFS得到的解是最优解，如果不成正比，得到的解不一定是最优解。

Time complexity: 1 + b + b2 +    + bd = O(bd ).

b为树的枝杈数，d为树的深度，若要搜索每个节点则需要

1 + b + b2 +    + bd = O(bd )的时间复杂度

Or O(bd+1) if goal test is applied after expansion.

如果目标测试应用在扩展之后，时间复杂度为O(bd+1)。

Space complexity: O(bd ) (keeps every node in memory)

因为最深的一层节点数为bd，搜索时需要队列支持，所以空间复杂度为O(bd )。