

# 人工智能基础第二次编程 实验报告

自96 曲世远 2019011455

## 1.作业要求

我本次作业选择的是完成八皇后问题，本题我采用的是回溯搜索的算法，由于问题复杂度较低，可以不采用较为复杂的回溯算法，只采用最基础的类似于添加了约束条件的DFS算法即可实现目标。

## 2.算法实现

```
1 def dfs(b, point):
2     valid_points = b.get_possible_moves()
3     if len(valid_points) == 0:
4         if len(b.queen) == 8:
5             return True
6         return False
7     else:
8         for point in valid_points:
9             b.make_move(point)
10            if dfs(b, point) == True:
11                return True
12            b.remove_move(point)
```

以上函数为我实现本题的主要算法，主要思路就是在递归到全部点时判断是否满足约束条件，如果不满足约束条件则返回继续搜索。

在本题最开始的尝试中，我也试图使用最少剩余项和最多约束项的方式进行回溯搜索，不过进行了复杂度分析之后，我觉得对于本题来说，上述的方式的代价与收益差别不大，于是还是采用了最基础的DFS改进的方式完成，下附最少约束项的确定方式：

```
1 def valuable_line(points, valid_lines):
2     #count the points in each line
3     lines = [0] * 8
4     #return line
5     ret = 0
6     #return points
7     valid_points = []
8     value = 8
9     value_lines = [0] * 8
10    for point in points:
11        value_lines[point // 8] = 1
12        lines[point // 8] += 1
13    i = -1
14    for line in lines:
15        i += 1
16        if valid_lines[i] * value_lines[i] == 0:
17            continue
18        if line < value:
19            value = line
20            ret = i + 1
21    for point in points:
```

```
22         if point // 8 == ret - 1:  
23             valid_points.append(point)  
24     return ret, valid_points
```