



算法设计与分析

作业（九）

姓 名	熊恪峥
学 号	22920202204622
日 期	2022年5月8日
学 院	信息学院
课程名称	算法设计与分析

作业（九）

目录

1	题12.2	1
2	题12.3	1
3	题12.4	1
4	题12.5	1

1 题12.2

对棋盘建立一个坐标系，则两个皇后在一个对角线上当且仅当 $k_{ij} = 1$ 或 $k_{ij} = -1$ 。即

$$k_{ij} = \frac{x_i - x_j}{i - j} = 1$$

$$\Rightarrow x_i - x_j = i - j$$

同理可得 $x_i - x_j = j - i$

2 题12.3

递归地对第1, 2, ..., 8行枚举放的位置，放棋子时检查同一行、同一列、对角线上是否有棋子。对角线按第1节的结论进行检查得到算法1。

算法 1 8皇后问题

```

1: procedure PUTQUENE( $k$ )
2:   if  $k = 8$  then
3:     PRINT( $pos$ )
4:   for  $i = 1 \rightarrow 8$  do
5:     if CHECK( $k, i$ ) then
6:        $pos[k] \leftarrow i$ 
7:        $t = \text{PUTQUENE}(k + 1)$ 
8:        $pos[k] \leftarrow 0$ 
9:   return  $t$ 
10: procedure CHECK( $row, col$ )
11:    $r = pos[row] == 0$ 
12:    $c = \text{True}$ 
13:   for  $i = 1 \rightarrow 8$  do
14:     if  $pos[i] == col$  then
15:        $c = \text{False}$ 
16:     break
17:    $d = \text{True}$ 
18:   for  $i = 1 \rightarrow 8$  do
19:     for  $j = 1 \rightarrow 8$  do
20:       if  $row - i == col - j$  or  $row - i = j - col$  then
21:         if  $pos[i] == j$  then
22:            $d = \text{False}$ 
23:         break
24:   return  $c$  and  $r$  and  $d$ 

```

3 题12.4

递归地检查所有棋子，然后标记所有能攻击的位置。最后检查是否所有的位置都被标记过了。如算法3

4 题12.5

枚举每个位置的放置方法，如算法??。

算法 2 检查能否攻击

```

1: procedure CHECK( $pos, n$ )
2:    $board[] = \emptyset$ 
3:   MARK( $board, pos, 0, n$ )
4:   for  $i = 1 \rightarrow n$  do
5:     for  $j = 1 \rightarrow n$  do
6:       if  $board[i][j] == 0$  then
7:         return False
8:   return True
9: procedure MARK( $board, pos, k, n$ )
10:   $r = pos[row] == 0$ 
11:   $c = True$ 
12:  for  $i = 1 \rightarrow 8$  do
13:    if  $pos[i] == col$  then
14:       $c = False$ 
15:      break
16:   $d = True$ 
17:  for  $i = 1 \rightarrow 8$  do
18:    for  $j = 1 \rightarrow 8$  do
19:      if  $row - i == col - j$  or  $row - i = j - col$  then
20:        if  $pos[i] == j$  then
21:           $d = False$ 
22:          break
23:  return  $c$  and  $r$  and  $d$ 

```

算法 3 检查能否攻击

```

1: procedure PERMU( $p, k, n$ )
2:   if  $k == n$  then
3:     PRINT( $p$ )
4:   for  $i = 1 \rightarrow n$  do
5:     if  $vis[i] = True$  then
6:        $vis[i] = True$ 
7:        $p[k] = i$ 
8:       PERMU( $p, k + 1, n$ )
9:        $vis[i] = False$ 

```
