**第二次会议**

1.确定团队名称：

异维突触实验室

2.论文分享：

李强：自然语言处理

张巧：计算机图像处理

徐聪：网络爬虫综述

3.算法讨论：

八皇后问题

**简介：**八皇后问题，是一个古老而著名的问题，是回溯算法的典型案例。该问题是国际西洋棋棋手马克斯·贝瑟尔于1848年提出。

**要求：**在8×8格的国际象棋上摆放八个皇后，使其不能互相攻击，即任意两个皇后都不能处于同一行、同一列或同一斜线上，问有多少种摆法。

**思想：**使用一维数组或二维数组来实现，因为必然八个皇后不在同一列同一行上，因此，每一行每一列必然存在一个皇后，所以，可以先固定皇后的行坐标，再使用递归回溯算法从第一行到最后一行依次遍历求出列坐标。列坐标需满足：皇后之间列坐标互易，且与位于前行的皇后不在同一对角线上。如果满足，则遍历下一行，如果不满足，则遍历同行下一列，如果都失败，则回溯到上一行的下一列，继续操作……如果最后一行的皇后确认，则输出序列，并继续回溯遍历，直至结束，输出种数。

**算法：**（from baidu）

**Python**

def queen(A, cur=0):

    if cur == len(A):

        print(A)

        return 0

    for col in range(len(A)):

        A[cur], flag = col, True

        for row in range(cur):

            if A[row] == col or abs(col - A[row]) == cur - row:

                flag = False

                break

        if flag:

            queen(A, cur+1)

queen([None]\*8)

**C++**

#include<iostream>

using namespace std;

static int gEightQueen[8] = { 0 }, gCount = 0;

void print()//输出每一种情况下棋盘中皇后的摆放情况

{

    for (int i = 0; i < 8; i++)

    {

        int inner;

        for (inner = 0; inner < gEightQueen[i]; inner++)

            cout << "0";

            cout <<"#";

        for (inner = gEightQueen[i] + 1; inner < 8; inner++)

            cout << "0";

        cout << endl;

    }

    cout << "==========================\n";

}

int check\_pos\_valid(int loop, int value)//检查是否存在有多个皇后在同一行/列/对//角线的情况

{

    int index;

    int data;

    for (index = 0; index < loop; index++)

    {

        data = gEightQueen[index];

        if (value == data)

            return 0;

        if ((index + data) == (loop + value))

            return 0;

        if ((index - data) == (loop - value))

            return 0;

    }

    return 1;

}

void eight\_queen(int index)

{

    int loop;

    for (loop = 0; loop < 8; loop++)

    {

        if (check\_pos\_valid(index, loop))

        {

            gEightQueen[index] = loop;

            if (7 == index)

            {

                gCount++, print();

                gEightQueen[index] = 0;

                return;

            }

            eight\_queen(index + 1);

            gEightQueen[index] = 0;

        }

    }

}

int main(int argc, char\*argv[])

{

    eight\_queen(0);

    cout << "total=" << gCount << endl;

    return 0;

}