**华南农业大学**

**《算法分析与设计》课程实验**

**实验8：八后问题**

年级专业： 21信息与计算科学

学生姓名： 吴嘉豪

学生学号： 202125810321

指导老师： 赵峰

实验时间： 2024年 05月29 日

一、实验内容

将回溯法改造为拉斯维加斯算法，解决八后问题，分析实验结果。

二、实验步骤

1、将回溯法改造为拉斯维加斯算法，求解八后问题，并编程实现；

2、对四后、五后问题求解，检验算法的正确性；

3、对八后问题，分别随机放置前个后。对于这9种情况，每种情况至少运行100次，并统计成功求出解的概率（记为），成功求出解时需要访问的结点数的平均值（记为），未能成功求出解时需要访问的结点数的平均值（记为）；

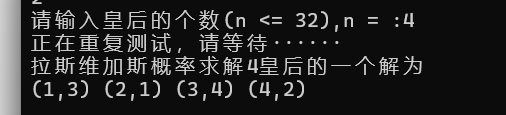
4、仿照教材P209的表7-1格式，制作实验结果分析表；

5、填写实验报告；

6、有兴趣的同学可以尝试求解12后问题，并将结果写入实验报告。

三、实验结果

2.



文本

描述已自动生成

四、源程序代码

#*include*<iostream>

#*include*<math.h>

#*include* <ctime>

#*include*<stdlib.h>

using namespace std;

int x[100],q[100];//*q[i]=n表示回溯解，x表示概率解*

int n,cnt = 0;

clock\_t start1, finish1;

clock\_t start2, finish2;

double totaltime1,totaltime2;

//回溯放置皇后时判断是否合法

bool find(int a[],int k)

{

    for(int i = 1;i < k;i++)

    {

        if(a[i] == a[k] || abs(k-i) == abs(a[k] - a[i]))

            return false;

    }

    return true;

}

bool Place(int x[],int k)

{

    for (int i = 0;i < k;i++)

        if(x[i] == x[k] || abs(i - k) == abs(x[i] - x[k]))

        {

            return true;

        }

    return false;

}

void print(int n)

{

    int i,j;

    cnt++;

    cout << "第 " << cnt << " 个解" << endl;

    for(i = 1;i <= n;i++)

        cout <<"(" <<  i << "," << q[i] << ")" << " ";//第i行放在第q[i]处。

    cout << endl;

    for(i = 1;i <= n;i++)        //行

    {

        for(j = 1;j <= n;j++)    //列

        {

            if(q[i] != j)

                printf("x ");

            else

                printf("Q ");

        }

        cout << endl;

    }

}

//回溯法求n皇后

void queen(int n){

    cnt = 0;

    for(int i = 0;i <= n;i++)//先将所有行的列数清零

    {

        q[i] = 0;

    }

    int k = 1;//k是行数

    while(k >= 1)

    {

        q[k] += 1;

        while(q[k] <= n )//未到达最后一列

        {

            if(find(q,k))

                break;

            else

                q[k]++;//该列不行向后推一个

        }

        if(k == n && q[k] <= n)//如果到最后一行，且最后列数正确

        {

            for(int i = 1;i <= n;i++)

            {

                cout << q[i] << " ";

            }

            cout << endl;

            print(n);//将皇后以棋盘式输出

        }

        else if(q[k] <= n)//如果还没有到达最后一列，继续下一列

        {

            k++;

        }

        else//回溯，因为到达最后一行，且列数遍历完，仍没有出结果

        {

            q[k] = 0;//让该行的列数重置为0

            k--;//继续回到上一行

        }

    }

}

//类，产生随机数

class RandomNumber{//类，这样才能保证不会出现产生的随机值都一样的情况。

public:

    RandomNumber(){

        srand(time(0));

    }

    int get(int begin = 0, int end = 1){//用这个函数可以得到随机值

        return rand()%(end-begin+1)+begin;

    }

};

bool Queen(int n) //拉斯维加斯概率求解

{

    RandomNumber arr[n + 1];

    int i,j,count=0;// //定义i,j,count. 其中count为试探次数

    for(i = 0;i < n;)// //其中 n为行数，for循环经过n次，每次代表在第n行中找到皇后所在的列

    {

            //为j赋值为一个随机数，此为拉斯维加斯概率算法的核心思想，通过随机性选择快速的引导算法

        j = arr[i + 1].get(1,n);//这样才可以保证每次的随机数都不一样，如果用j=rand()%n+1;不可以

        x[i]=j;  //将随机选取的j作为列向量参与接下来的运算

        //试探次数加一，用于确定N次运算后n列全部实验完毕

        count++;

        if(!Place(x,i))   //这个if语句用于调用冲突子函数以确定是否发生冲突，即这个位置是否可以放置皇后

        {

            count=0;

            i++;

        }

        else if(count==n)   //位置冲突

        {

            return false;

        }

    }

    cout << "拉斯维加斯概率求解" << n << "皇后的一个解为" << endl;

    for(i=0;i<n;i++)//  //作为输出结果时的for循环

    {

       cout <<"(" <<  i + 1 << "," << x[i] << ")" << " ";

    }

    cout << endl;

*return* true;

}

void *runtime*()

{

    totaltime1 = finish1 - start1;

    totaltime2 = finish2 - start2;

    cout << "回溯法求解"<< n << "皇后的运行时间为" << totaltime1 << "毫秒" << endl;

    cout << "拉斯维加斯概率算法求解 "<< n << "皇后的运行时间为" << totaltime2  << "毫秒"<< endl;

}

int *choose*()

{

    int ch;

    cout << endl << endl << endl;

    cout << "-----------------------n皇后问题---------------------" << endl;

    cout << "                请选择你要进行的操作                   " << endl;

    cout << "   1、回溯法求解              2、拉斯维加斯概率算法求解   " << endl;

    cout << "   3、查看两者运行时间比较              4、退出程序              " << endl;

    cout << "---------------------------------------------------" << endl;

    cin >> ch;

*return* ch;

}

int *main*()

{

    int ch = *choose*();

*while*(ch != 4)

    {

        switch (ch) {

            case 1:

                printf("请输入皇后的个数(n <= 32),n = :");

                cin >> n;

                start1 = clock();//开始计时

                queen(n);

                finish1 = clock();

                break;

            case 2:

                printf("请输入皇后的个数(n <= 32),n = :");

                cin >> n;

                start2 = clock();//开始计时

                cout << "正在重复测试，请等待······" << endl;

                while(!Queen(n))

                {

                }

                finish2 = clock();

                break;

            case 3:

                runtime();

                break;

            default:

                exit(0);

        }

        ch =  choose();

    }

    return 0;

}

五、评语及评分

|  |
| --- |
| 评语及评分  评阅人签名： |