**算法分析与设计实验报告**

**第 1 次实验**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 杨杰 | 学号 | 201908010705 | | 班级 | 计科1907 |
| 时间 | 3.25 | 地点 | 软件大楼 | | | |
| 实验名称 | 金币阵列问题 | | | | | |
| 实验目的 | 通过上机实验，要求掌握金币阵列问题的问题描述、算法设计思想、程序设计。 | | | | | |
| 实验原理 | 计算出金币阵列的最少变换次数，根据不同数据规模的测试用例，能准确输出测试用例的最少变换次数，并计算出程序运行所需要的时间。 | | | | | |
| 实验步骤 | 1. 将矩阵中的每一列作为第1列，并将第1列中的相关元素与目标矩阵中第1列的元素进行配对，如果不相同，则利用第1个规则进行翻转； 2. 从第2列开始，将处理后的列与目标列进行比较，如果相同，则转下一列；如果不相同，看是否可以通过列的交换完成，如果不可以，则无法做到，如果可以，则继续扫描，直至所有的列扫描完成为止。 | | | | | |
| 关键代码 | void trans1(int i,int n)*//对某一行进行翻转*  {      for(int j=0;j<n;j++)          temparr[i][j]=1-temparr[i][j];*// a[j][i]^=1; //异或运算也可以*      count++;  }  void trans2(int i,int j,int m)*//对两列进行互换*  {      if(i!=j)      {          for(int k=0;k<m;k++)          {              int temp=temparr[k][i];              temparr[k][i]=temparr[k][j];              temparr[k][j]=temp;          }          count++;      }  }  bool same(int i,int j,int m)*//判断两列是否相同*  {      int k;      bool flag=true;      for(k=0;k<m;k++)      {          if(temparr[k][j]!=b[k][i])          {              flag=false;              break;          }      }      return flag;  }  for(k=0;k<m;k++)          {  *//copyarray*              for(i=0;i<m;i++)                  for(j=0;j<n;j++)                      temparr[i][j]=a[i][j];              count=0;              trans2(0,k,m);*//将第k列与第0列互换*  *//对所有行的第0个元素与目标数组对应的元素进行比较，如果不同,进行行翻转*              for(i=0;i<m;i++)                  if(temparr[i][0]!=b[i][0])                      trans1(i,n);  *//检查每一列是否满足条件*              bool found;              for(i=0;i<n;i++)              {                  found=false;                  if(same(i,i,m))                  {                      found=true;                      continue;                  }                  for(j=i+1;j<n;j++)*//判断是否可以通过后续列的交换达到目的*                  {                      if(same(i,j,m))                      {                          trans2(i,j,m);                          found=true;                          break;                      }                  }                  if(found==false)                      break;              }              if(found==true && count<answer) answer=count;          } | | | | | |
| 算法复杂度分析 | 先逐个函数进行分析，trans1函数一层for循环，时间复杂度为O(n);trans2函数一层for循环，时间复杂度为O(m); same函数一层for循环，时间复杂度为O(m);  在main函数中，三层for循环，加上same函数和trans2函数，总时间复杂度为O(m3n2). | | | | | |
| 测试结果  （含运行时间） |  | | | | | |
| 实验心得 | 觉得正确但是错误的一种思路：  从0号行开始判断，如果coin1和coin2的某一行的1的个数相加等于col的大小，则翻转；如果coin1某一行的1的个数等于coin2的某一行的1的个数，则不需要翻转；如果coin1和coin2的某一行的1的个数既不相等，相加也不等于col的大小，则无法翻转。  此种分析看似合理，代码也能写出来，但是col为偶数时，此程序会混乱，col为奇数，则能正常执行。  本题生成的随机数据很少有能满足要求的，这启示我们数据的制造也要讲究一些方法、策略，本题是否有更好的数据生成方法仍有待发掘。  经过这次实验，我对于金币阵列的相关代码已基本熟悉，算法知识得到了复习与巩固。在写代码与调试的过程中，在解决问题过程中，丰富了个人编程的经历和经验，提高了个人解决问题的能力。 | | | | | |
| 实验得分 |  | 助教签名 | |  | | |

**附录：完整代码**

*//金币阵列问题*

#include <iostream>

#include <fstream>

#include <windows.h>

using namespace std;

int count=0;

int temparr[100][100],a[100][100],b[100][100];

void trans1(int i,int n)*//对某一行进行翻转*

{

    for(int j=0;j<n;j++)

        temparr[i][j]=1-temparr[i][j];*// a[j][i]^=1; //异或运算也可以*

    count++;

}

void trans2(int i,int j,int m)*//对两列进行互换*

{

    if(i!=j)

    {

        for(int k=0;k<m;k++)

        {

            int temp=temparr[k][i];

            temparr[k][i]=temparr[k][j];

            temparr[k][j]=temp;

        }

        count++;

    }

}

bool same(int i,int j,int m)*//判断两列是否相同*

{

    int k;

    bool flag=true;

    for(k=0;k<m;k++)

    {

        if(temparr[k][j]!=b[k][i])

        {

            flag=false;

            break;

        }

    }

    return flag;

}

int main()

{

    ifstream infile("jinbi\_in.txt",ios::in);

    ofstream outfile("jinbi\_out.txt",ios::out);

    int t=0;

    infile>>t;

    int m=0,n=0,i,j,k;

    double time=0;

    LARGE\_INTEGER nFreq,nBeginTime,nEndTime;

    while(infile>>m>>n)

    {

        int answer=9999;

        QueryPerformanceFrequency(&nFreq);

        QueryPerformanceCounter(&nBeginTime);*//开始计时*

        for(i=0;i<m;i++)

            for(j=0;j<n;j++)

                infile>>a[i][j];

        for(i=0;i<m;i++)

            for(j=0;j<n;j++)

                infile>>b[i][j];

        for(k=0;k<m;k++)

        {

*//copyarray*

            for(i=0;i<m;i++)

                for(j=0;j<n;j++)

                    temparr[i][j]=a[i][j];

            count=0;

            trans2(0,k,m);*//将第k列与第0列互换*

*//对所有行的第0个元素与目标数组对应的元素进行比较，如果不同,进行行翻转*

            for(i=0;i<m;i++)

                if(temparr[i][0]!=b[i][0])

                    trans1(i,n);

*//检查每一列是否满足条件*

            bool found;

            for(i=0;i<n;i++)

            {

                found=false;

                if(same(i,i,m))

                {

                    found=true;

                    continue;

                }

                for(j=i+1;j<n;j++)*//判断是否可以通过后续列的交换达到目的*

                {

                    if(same(i,j,m))

                    {

                        trans2(i,j,m);

                        found=true;

                        break;

                    }

                }

                if(found==false)

                    break;

            }

            if(found==true && count<answer) answer=count;

        }

        QueryPerformanceCounter(&nEndTime);*//停止计时*

        time=(double)(nEndTime.QuadPart-nBeginTime.QuadPart)/(double)nFreq.QuadPart;*//计算程序执行时间单位为ms*

        if(answer<9999)

            outfile<<"程序耗时"<<time\*1000<<"ms,结果为"<<answer<<endl;

        else

            outfile<<"程序耗时"<<time\*1000<<"ms,结果为"<<-1<<endl;

    }

*//整体思路，将每一列运用列交换作为第0列，然后对每一行进行判断，如果元素与目标元素不相等,进行翻转处理，再向后判断相关列是否相同。*

    infile.close();

    outfile.close();

    return 0;

}

*//金币阵列数据生成器*

#include <iostream>

#include <fstream>

#include <time.h>

#define random(a,b) (rand()%(b-a+1)+a)

using namespace std;

int main()

{

    int t=0;

    cin>>t;

    ofstream outfile("jinbi\_in.txt",ios::out);

    outfile<<t<<endl;

    srand((int)time(NULL));

    int m=0,n=0;

    for(int i=0;i<t;i++)

    {

        m=random(2,100);

        n=random(2,100);

        outfile<<m<<' '<<n<<endl;

        for(int j=0;j<m;j++)

        {

            for(int k=0;k<n;k++)

                outfile<<random(0,1)<<' ';

            outfile<<endl;

        }

        for(int j=0;j<m;j++)

        {

            for(int k=0;k<n;k++)

                outfile<<random(0,1)<<' ';

            outfile<<endl;

        }

}

outfile.close();

    return 0;

}