**《算法设计与分析》实验报告**

# 实验四 回溯法

|  |  |  |  |
| --- | --- | --- | --- |
| **报告书** | | | |
| 姓名 | 吴宇敖 | 指导教师 |  |
| 学号 | 176001752 | 日 期 | 2020.5.16 |
| 班级 | 17计算机软件一班 |  |  |
| **实验内容** | | | |
| **[要求先在OJ上提交通过]**  请给出AC截图和代码；无需说明。  对查重结果问题请在文档开头予以说明并给出前后源码。  1）森林迷宫  2）地图着色 | | | |
| **实验目的** | | | |
| 堂练目的：考察学生对本阶段知识点及编程掌握的熟练程度 | | | |
| **实验过程和步骤** | | | |
| **【实验题目】**  **题目描述：**  **一天Luna在森林里探险的时候不小心走入了一个迷宫，迷宫可以看成是由n \* n的格点组成，每个格点只有2种状态：^ 和 # ；前者表示可以通行后者表示不能通行。当Luna处在某个格点时，她只能移动到东南西北(或者说上下左右)四个方向之一的相邻格点上，Luna想要从起点A走到终点B（中途不能走出迷宫）。如果起点或者终点有一个不能通行(为#)，则当做无法通行。**  **输入要求：**  **第1行是测试数据的组数k，后面跟着k组输入。**  **每组测试数据的第1行是一个正整数n (1 <= n <= 100)，表示迷宫的规模是n \* n的。**  **接下来是一个n \* n的矩阵，矩阵中的元素为 . 或者 #。**  **再接下来一行是4个整数ar,ac,br,bc。表示A处在第ar行，第ac列，B处在第br行, 第bc列。注意坐标ar，ac，br，bc全部是从0开始计数的**  **输出要求：**  **YES或NO**  **【程序代码及注释】**  #include<iostream>  using namespace std;  int num[100][100];  bool cmp;  bool find(int x1,int y1,int x2,int y2,int n)//寻找路径  {  if(x1==x2&&y1==y2)//到达终点后返回  return cmp=true;  if(num[x1][y1]==1||num[x2][y2]==1)//起点或终点不可走  return cmp=false;  if(num[x1][y1]==2)//防止进入死循环  num[x1][y1]=0;  if(x1!=0&&num[x1-1][y1]==0)//向上  {  num[x1][y1]=2;//防止进入死循环  find(x1-1,y1,x2,y2,n);  }  if(x1!=n-1&&num[x1+1][y1]==0)//向下  {  num[x1][y1]=2;  find(x1+1,y1,x2,y2,n);  }  if(y1!=0&&num[x1][y1-1]==0)//向左  {  num[x1][y1]=2;  find(x1,y1-1,x2,y2,n);  }  if(y1!=n-1&&num[x1][y1+1]==0)//向右  {  num[x1][y1]=2;  find(x1,y1+1,x2,y2,n);  }      }  int main()  {  int n,m,ar,ac,br,bc;  char x;  cin>>m;  for(int k=0;k<m;k++)//初始化迷宫  {  cin>>n;  for(int i=0;i<n;i++)  for(int j=0;j<n;j++)  {  cin>>x;  if(x=='#')  num[i][j]=1;  else if(x=='^')  num[i][j]=0;  }    cin>>ar>>ac>>br>>bc;//输入起点和终点  cmp=false;  find(ar,ac,br,bc,n);  if(cmp)  cout<<"YES"<<endl;  else cout<<"NO"<<endl;  }  //system("pause");  return 0;  }  **【实验结果】**      **【实验题目】**  **题目描述：**  **给定图G=(V, E)，需要为图G的各顶点着色，是否有一种着色法使G中相邻的两个顶点有不同的颜色?**  **输入要求：**  **第一行是顶点的个数n（2≤n≤100），颜色数m（1≤m≤n）。**  **接下来是顶点之间的连接关系：a b；表示a和b相邻。顶点从1开始计。**  **当a，b同时为0时表示输入结束。**  **输出要求：**  **输出着色方案总数和最少颜色数。如果无可行方案，颜色数为0。**  **【程序代码及注释】**  #include<iostream>  using namespace std;  int x[100]={0};  int num[100][100]={0};  int sum=0;//一共有几种  int n1=0;//最少的颜色数  void color(int k,int m,int n)  {  if(k>m)//涂完后，退出  {  return ;  }  int col[100]={0};//记录第几种颜色，是否已经使用，帮助记录最少的颜色数  int n2=0;  for(int i=1;i<=n;i++)  {  x[k]=i;  int f=0;  for(int j=1;j<=m;j++)//每个颜色开始循环  if(num[k][j]==1&&x[k]==x[j])//旁边的两点相接，并且颜色相同  f=1;  if(f==0)  {  if(k==m)//当达到最后的一个点时，判断最少的颜色数  {  sum++;  for(int l=1;l<=m;l++)  if(col[x[l]]==0)  {  col[x[l]]=1;  n2++;  }  if(n2<n1)  n1=n2;  }  color(k+1,m,n);//进行下一个点  }  }  x[k]=0;  }  int main()  {  int m,n,x,y;  cin>>m>>n;  n1=n;  while(1)//记录哪两个顶点之间有连接  {  cin>>x>>y;  if(x==0&&y==0)  break;  num[x][y]=1;  num[y][x]=1;  }  color(1,m,n);//从第一个顶点开始  if(sum!=0)  cout<<sum<<' '<<n1<<endl;  else//没有可以达成条件的涂法  {  n1=0;  cout<<sum<<' '<<n1<<endl;  }  //system("pause");  return 0;  }  **【实验结果】** | | | |