分支界限法：输入电路板区域n\*m以及布线的起始位置和结束位置，输出布线方案

实验代码：

#include <iostream>

    #include<stdio.h>

    using namespace std;

    typedef struct

    {

    int row ;

    int col ;

    }Position;

    typedef struct

    {

    int row[10000] ;

    int col[10000] ;

   int end;

    int begin ;

    }Queue;

    int grid[100][100];

    Position start, finish;

    int PathLen = 0;

    Position \* path;

    int n , m , a , b , x ;

    bool FindPath(Position start,Position finish)

    {

    if((start.row==finish.row) && (start.col==finish.col))

    {

    PathLen=0;

    return true;

    }

    int i ;

    for( i=0; i<= m+1; i++)

    grid[0][i]=grid[n+1][i]=1;

    for( i=0; i<= n+1; i++)

    grid[i][0]=grid[i][m+1]=1;

    int j ;

    Position offset[4];

    offset[0].row=0; offset[0].col=1;//右

    offset[1].row=1; offset[1].col=0;//下

    offset[2].row=0; offset[2].col=-1;//左

    offset[3].row=-1; offset[3].col=0;//上

    int NumOfNbrs=4;

    Position here,nbr;

    here.row=start.row;

    here.col=start.col;

    grid[start.row][start.col]=2;

    Queue Q ;

    Q.end = 0 ;

    Q.begin = 0 ;

    do {

    for( i=0; i<NumOfNbrs; i++)

    {

    nbr.row=here.row + offset[i].row;

    nbr.col=here.col+offset[i].col;

    if(grid[nbr.row][nbr.col]==0)

    {

    grid[nbr.row][nbr.col]=grid[here.row][here.col]+1;

    if((nbr.row==finish.row) &&(nbr.col==finish.col))

    break;

    Q.col[Q.end] = nbr.col;

    Q.row[Q.end] = nbr.row;

    Q.end++;

    }

    }

    if((nbr.row==finish.row)&&(nbr.col==finish.col)) break;

    if(Q.begin==Q.end) return false;

    else

    {

    here.row=Q.row[Q.begin];

    here.col=Q.col[Q.begin];

    Q.begin++;

    }

    }while(true);

    PathLen=grid[finish.row][finish.col]-2;

    path=new Position[PathLen];

    here=finish;

    for( j = PathLen-1; j>=0; j--){

    path[j]=here;

    for( i=0; i<NumOfNbrs; i++){

    nbr.row=here.row+offset[i].row;

    nbr.col=here.col+offset[i].col;

    if(grid[nbr.row][nbr.col]==j+2) break;

    }

    here=nbr;

    }

    return true;

    }

    int main()

    {

    int j ;

    printf("输入电路板区域n\*m和封锁的格数x:\n");

    cin>>n>>m>>x;

    printf("输入封锁的坐标:\n");

    for( a = 0 ; a < n+2 ; a ++ )

    for( b = 0 ; b < m +2 ; b ++ )

    grid[a][b] = 0 ;

    for( x = x ; x >= 1 ; x -- )

    {

    cin >> a >> b ;

    grid[a][b]= 1;

    }

    printf("输入起始位置和结束位置:\n");

    cin>>start.row>>start.col>>finish.row>>finish.col;

    if(FindPath(start,finish))

    {

    printf("输出路径长度及途中坐标:");

    cout<<PathLen<<endl;

    cout<<start.row<<" "<< start.col<<endl;

    for( j = 0 ; j < PathLen ; j++ )

    cout<<path[j].row<<" "<<path[j].col<<endl;

    }

    else cout<<"No Solution!"<<endl;

    delete []path;

    return 0 ;

    }