

# 深度优先搜索

1.....void dfs(int step)

{

    尝试边界

    尝试每一种可能for ( int i=1;i<=n;i++ )

    {

        继续下一步dfs(step+1);

    }

}

2...../\*求在迷宫内，一个点到另外一个点的最短路径\*/

#include<iostream>

using namespace std;

int a[51][51],book[51][51],p,q,n,m;

int mi=99999;

void dfs(int x,int y,int step){

    int next[4][2]={{0,1},{1,0},{0,-1},{-1,0}};//分别表示方向，右下左上

上

    if(x==p && y==q){//判断是否到达终点（边界位置）

        if(step<mi)

        mi=step;

```

        return ;
    }
    for(int k=0;k<=3;k++){//枚举4种走法（上下左右）
        int tx=x+next[k][0];
        int ty=y+next[k][1];
        if(tx<1||tx>n||ty<1||ty>m)//判断是否数组越界
            continue;
        if(a[tx][ty]==0&&book[tx][ty]==0){//判断是否障碍物或者已经
            在路径中
                book[tx][ty]=1;
                dfs(tx,ty,step+1);//尝试下一个点
                book[tx][ty]=0;
            }
        }
    }
}

int main(){
    int i,j;
    int stx,sty;
    cout<<"请输入n,m"<<endl;
    cin>>n>>m;
    for(i=1;i<=n;i++)
        for(j=1;j<=m;j++){
            cin>>a[i][j];
        }
}

```

```

    cout<<"请输入起点和终点"<<endl;
    cin>>stx>>sty>>p>>q;
    book[stx][sty]=1;
    dfs(stx,sty,0);
    cout<<"最短路径为"<<mi;
    return 0;
}

```

3...../\*判断 $XXX+XXX=XXX$ ,要求各个X是1~9的不同数字\*/

```

#include<iostream>
using namespace std;
int a[10],book[10],total;
void dfs(int step){//表示站在第几个盒子前面
    int i;
    if(step==10){          //如果站在第10个盒子前面，则表示前面9
个已经放好扑克牌
        int x=100*a[1]+10*a[2]+a[3];
        int y=100*a[4]+10*a[5]+a[6];
        int z=100*a[7]+10*a[8]+a[9];
        if(x+y==z){
            total++;
            cout<<x<<"+"<<y<<"="<<z<<endl;
        }
        return;
    }
}

```

```

    }
    for(int i=1;i<=9;i++){
        if(book[i]==0){    //判断扑克牌book[i]是否在手上 , 0
            则在
                a[step]=i;
                book[i]=1;
                dfs(step+1);
                book[i]=0 ;    //将刚刚尝试的扑克牌收回 , 才能
            进行下一次尝试
        }
    }
    return ;
}
int main(){
    dfs(1);
    cout<<"total="<<total/2<<endl;
    return 0;
}

```