迭代加深算法IDA

题目描述：

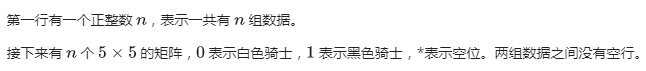
在一个5×5的棋盘上有12个白色的骑士和12个黑色的骑士，且有一个空位。在任何时候一个骑士都能按照骑士的走法（它可以走到和它横坐标相差为1，纵坐标相差为2或者横坐标相差为2，纵坐标相差为1的格子）移动到空位上。

给定一个初始的棋盘，怎样才能经过移动变成如下目标棋盘：



为了体现出骑士精神，他们必须以最少的步数完成任务。

输入：



输出：

IMG_256

输入样例：

2

10110

01\*11

10111

01001

00000

01011

110\*1

01110

01010

00100

输出样例：

7

-1

思路分析:

根据迭代加深搜索，可以将移动步数为搜索深度，然后步步搜索，然后我们可以用一个目标数组置为答案。

之后，再用一个乐观估计函数做剪枝，但是可是这该如何定义呢？

可以设想，假设现在的数组与目标数组的相差值与接下来的可用步数大，

那么不可能得到答案，所以可以返回。

代码实现;

#include<cstdio>

#include<iostream>

#include<cstring>

using namespace std;

int find1=0,dx[9]={0,1,1,-1,-1,2,2,-2,-2},dy[9]={0,2,-2,2,-2,1,-1,1,-1},t,a[10][10],sx,sy;

char w;

int goal[5][5]={

{1,1,1,1,1},

{0,1,1,1,1},

{0,0,2,1,1},

{0,0,0,0,1},

{0,0,0,0,0}

};

int check()

{

int dif=0;

for(int i=0;i<5;i++)

for(int j=0;j<5;j++)

{

if(goal[i][j]!=a[i][j])

dif++;

}

return dif;

}

void iddfs(int x,int y,int de,int k)

{

if(de==k)

{

if(!check())

find1=1;

return;

}

for(int i=1;i<=8;i++)

{

int x1=x+dx[i];

int y1=y+dy[i];

if(x1<0||x1>4||y1<0||y1>4)

continue;

swap(a[x][y],a[x1][y1]);

if(de+check()<=k)

iddfs(x1,y1,de+1,k);

swap(a[x][y],a[x1][y1]);

}

}

int main()

{

scanf("%d",&t);

while(t--)

{

for(int i=0;i<5;i++){

getchar();

for(int j=0;j<5;j++)

{

scanf("%c",&w);

if(w=='\*')

{

a[i][j]=2;

sx=i;

sy=j;

}

else

a[i][j]=w-'0';

}

}

int k;

for(k=0;k<=15;k++)

{

iddfs(sx,sy,0,k);

if(find1)

break;

}

if(k>15)

printf("-1\n");

else

printf("%d\n",k);

find1=0;

}

}

