五子棋中极大极小搜索算法

int minMax(int co[20][20],int deep,int a,int Alpha,int Beta)

//分数传递,a为1表示白棋，为2表示黑棋,调用时Alpha，Beta赋为NGIF,PTIF

{

int i,j;

int c[20][20];

int minmax;

int b;

int n=1;

Tree tree;

tree.data=NGIF;

tree.Alpha=Alpha;

tree.Beta=Beta;

tree.X=0;

tree.Y=0;

if(a==1)

b=2;

else

b=1;

if(deep>0)

{

for(i=0;i<20;i++)

for(j=0;j<20;j++)

{

if(co[i][j]==0&&jdgen(co,i,j,1)&&tree.Alpha<tree.Beta)//α-β剪枝

{

memcpy(c,co,sizeof(int)\*400);

c[i][j]=a;

minmax=minMax(c,deep-1,b,tree.Alpha,tree.Beta);//递归调用

if(deep%2==0)

{

if(n==1)

{

tree.data=minmax;

n++;

}

if(tree.Alpha<minmax)

{

tree.Alpha=minmax;

tree.data=minmax;

tree.X=i;

tree.Y=j;

}

}

else

{

if(n==1)

{

tree.data=minmax;

n++;

}

if(tree.Beta>minmax)

{

tree.Beta=minmax;

tree.data=minmax;

tree.X=i;

tree.Y=j;

}

}

}

}

for(i=0;i<20;i++)

for(j=0;j<20;j++)

{

if(co[i][j]==0&&jdgen(co,i,j,2)&&tree.Alpha<tree.Beta)//α-β剪枝

{

memcpy(c,co,sizeof(int)\*400);

c[i][j]=a;

minmax=minMax(c,deep-1,b,tree.Alpha,tree.Beta);//递归调用

if(deep%2==0)

{

if(tree.Alpha<minmax)

{

tree.Alpha=minmax;

tree.data=minmax;

tree.X=i;

tree.Y=j;

}

}

else

{

if(tree.Beta>minmax)

{

tree.Beta=minmax;

tree.data=minmax;

tree.X=i;

tree.Y=j;

}

}

}

}

X=tree.X;

Y=tree.Y;

return tree.data;

}

else

{

return score(a,co,1);//局面评分

}

}