5-17

算法分析：

本题是子集树，从上到下依次搜索，暴力搜，还没想好剪枝方案。

求老师上课解答。

#include<iostream>

#include<cstring>

using namespace std;

int board[100][100];//房间被监视的情况

int robot[100][100];//机器人位置

int best[100][100];//最优解位置

int m,n;

int robotnum=0;//机器人个数

int bestknum=1<<20;//机器人个数最优解

int t=0;//被监视房间个数

//int t1,t2,more;

int dx[5]={0,1,-1,0,0};

int dy[5]={0,0,0,1,-1};

void change(int i,int j)//[i,j]处放置机器人

{

robot[i][j]=1;

robotnum++;

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

{

int p=i+dx[d];

int q=j+dy[d];

board[p][q]++;//因为可重复监视，所以是++,如果等于1的话，那么重复监视会错

if(board[p][q]==1)//从0变成1才房间数+1

t++;

}

}

void reset(int i,int j)//回溯法恢复监视前状态

{

robot[i][j]=0;

robotnum--;

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

{

int p=i+dx[d];

int q=j+dy[d];

board[p][q]--;

if(board[p][q]==0)

t--;

}

}

void backtrack(int i,int j)

{

do{

j++;

if(j>n)

{

i++;

j=1;

}

}while(board[i][j]>0&&i<=n);

if(i>n)//到达叶子节点

{

if(robotnum<bestknum)

{

bestknum=robotnum;

for(int x=1;x<=m;x++)

for(int y=1;y<=n;y++)

best[x][y]=robot[x][y];

return;

}

}

if(i<n)

{

change(i+1,j);

backtrack(i,j);

reset(i+1,j);

}

if(j<m&&(board[i][j+1]==0||board[i][j+2]==0))

{

change(i,j+1);

backtrack(i,j);

reset(i,j+1);

}

if(board[i+1][j]==0&&board[i][j+1]==0)

{ change(i,j);

backtrack(i,j);

reset(i,j);

}

}

int main()

{

cin>>m>>n;

if(m==1&&n==1)

{

cout<<1<<endl<<1<<endl;

}

else{

memset(board,0,sizeof(board));

memset(robot,0,sizeof(robot));

for(int i=0;i<=n+1;i++)

{

board[i][m+1]=1;

board[i][0]=1;

}

for(int i=0;i<=m+1;i++)

{

board[n+1][i]=1;

board[0][i]=1;

}

backtrack(1,0);

cout<<bestknum<<endl;

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

{ for(int j=1;j<=n;j++)

cout<<best[i][j]<<" ";

cout<<endl;

}

}

return 0;

}

