**dfs:**

**POJ3009**

void dfs(int posX,int posY){

if(steps>=10)

return;

for(int i=0;i<4;i++){

int tmpX = posX;

int tmpY = posY;

while(1){

tmpX += add[i][0];

tmpY += add[i][1];

if(tmpX<0||tmpX>=row||tmpY<0||tmpY>=col)

break;

if(tmpX==endX&&tmpY==endY){

steps++;

if(mini>steps)

mini = steps;

steps--;

return;

}

else if(arr[tmpX][tmpY]==1){//碰到障碍物

if(tmpX-add[i][0]!=posX||tmpY-add[i][1]!=posY){

arr[tmpX][tmpY] = 0;

steps++;

dfs(tmpX-add[i][0],tmpY-add[i][1]);

arr[tmpX][tmpY] = 1;

steps--;

}

break;

}

}

}

}

**bfs:**

**POJ3669**

int bfs()

{

memset(visited,0,sizeof(visited));

queue<P> que;

P current;

current.x = 0;

current.y = 0;

//当前花费时间

current.t = 0;

que.push(current);

while(que.size()){

//做个备份

const P p = que.front();que.pop();

for(int i=0;i<4;i++){

current = p;

current.x += direction[i][0];

current.y += direction[i][1];

++current.t;

if(current.x>=0&&current.y>=0&&map[current.x][current.y]>current.t&&!visited[current.x][current.y]){

visited[current.x][current.y] = true;

//爆炸时间大于当前时间，是安全的

if(map[current.x][current.y]>last)

//当前位置爆炸时间大于流星雨最晚落下的时间，说明跑出了流星雨区域

return current.t;

que.push(current);

}

}

}

return -1;

}