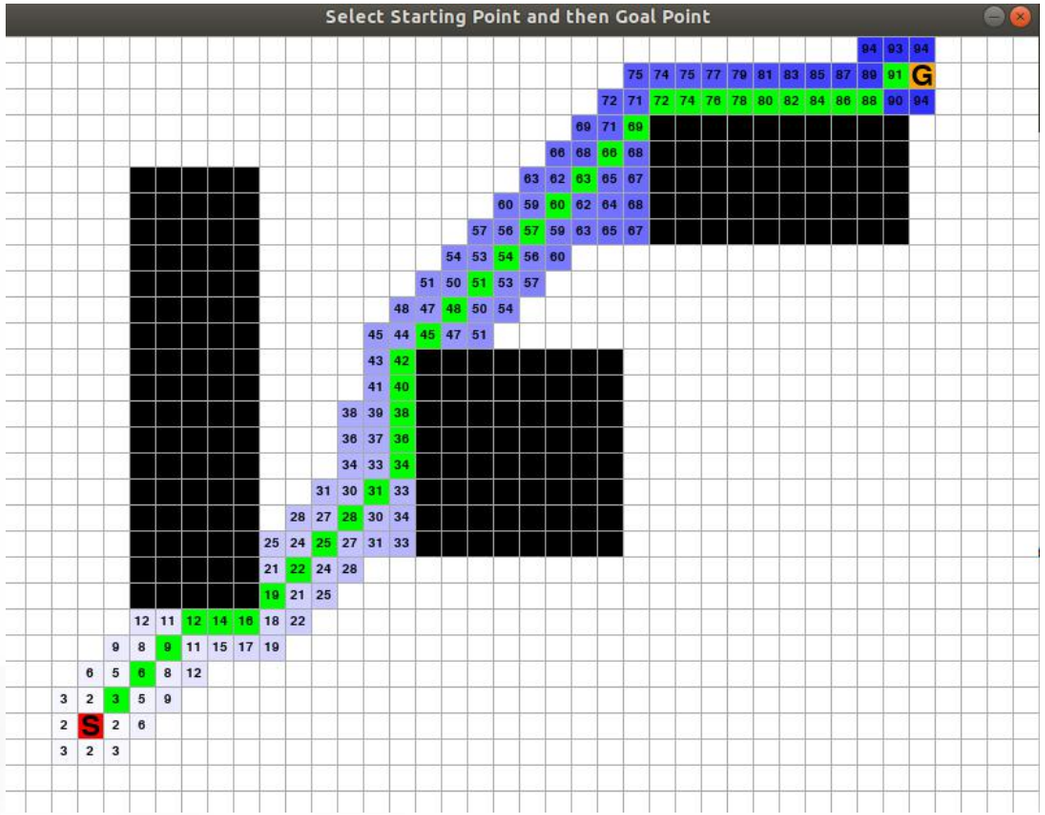
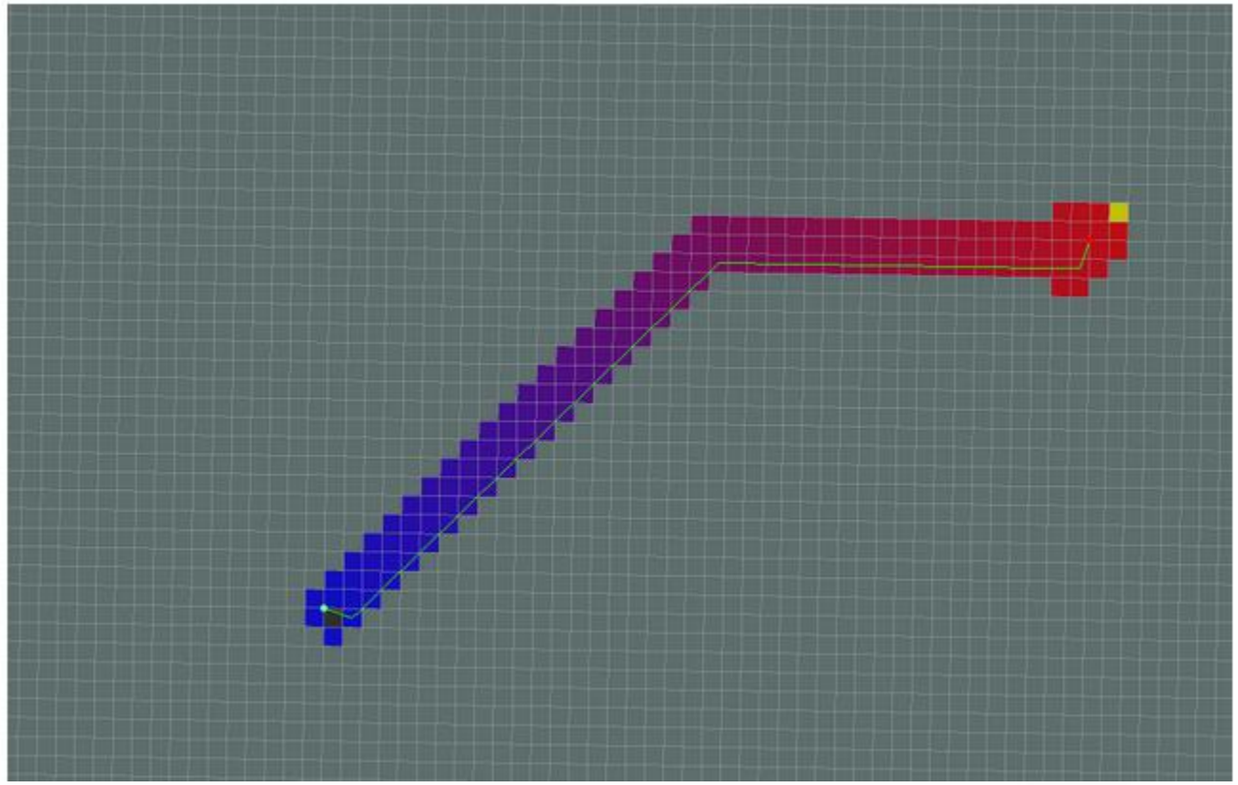
WHEELTEC
轮 趣 科 技

全局路径规划
——A*算法

A*算法效果图——深度优先

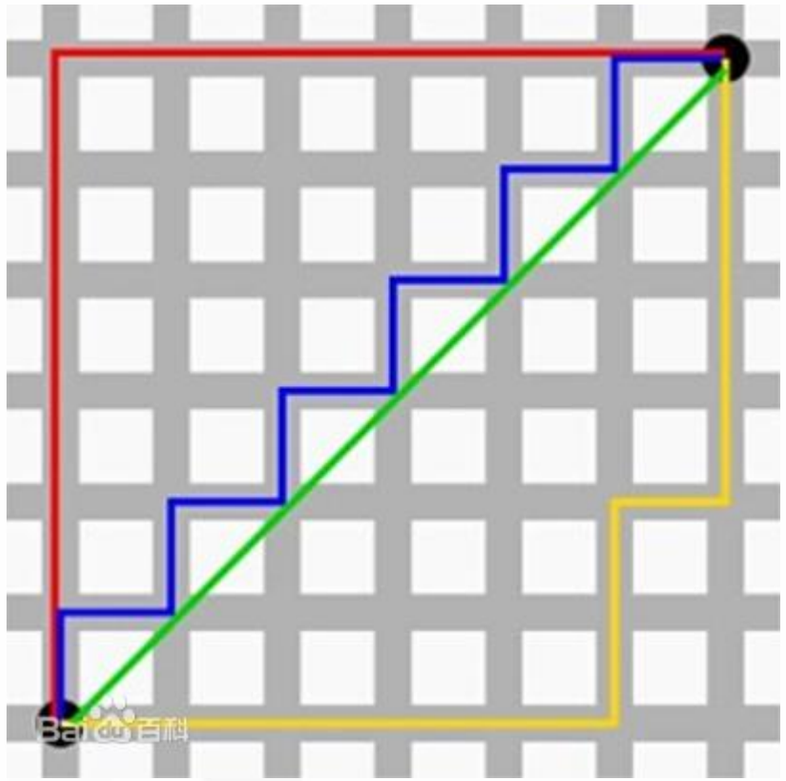




需要下载pygame

Ubuntu： sudo apt-get install python-pygame

两种距离——欧几里得距离、曼哈顿距离



绿色，代表欧几里得距离：

$$d(a,b)=\sqrt{(a_x-b_x)^2+(a_y-b_y)^2}$$

红色、蓝色、黄色，代表曼哈顿距离：

$$d(a,b)=|a_x-b_x|+|a_y-b_y|$$

红黄蓝距离等效，曼哈顿距离也称出租车距离

A*算法关键：估算函数

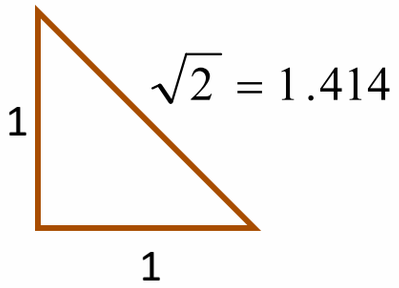
F代表从起点到终点估算路程

$$F^* = G + H^*$$

G代表从起点到当前节点实际路程

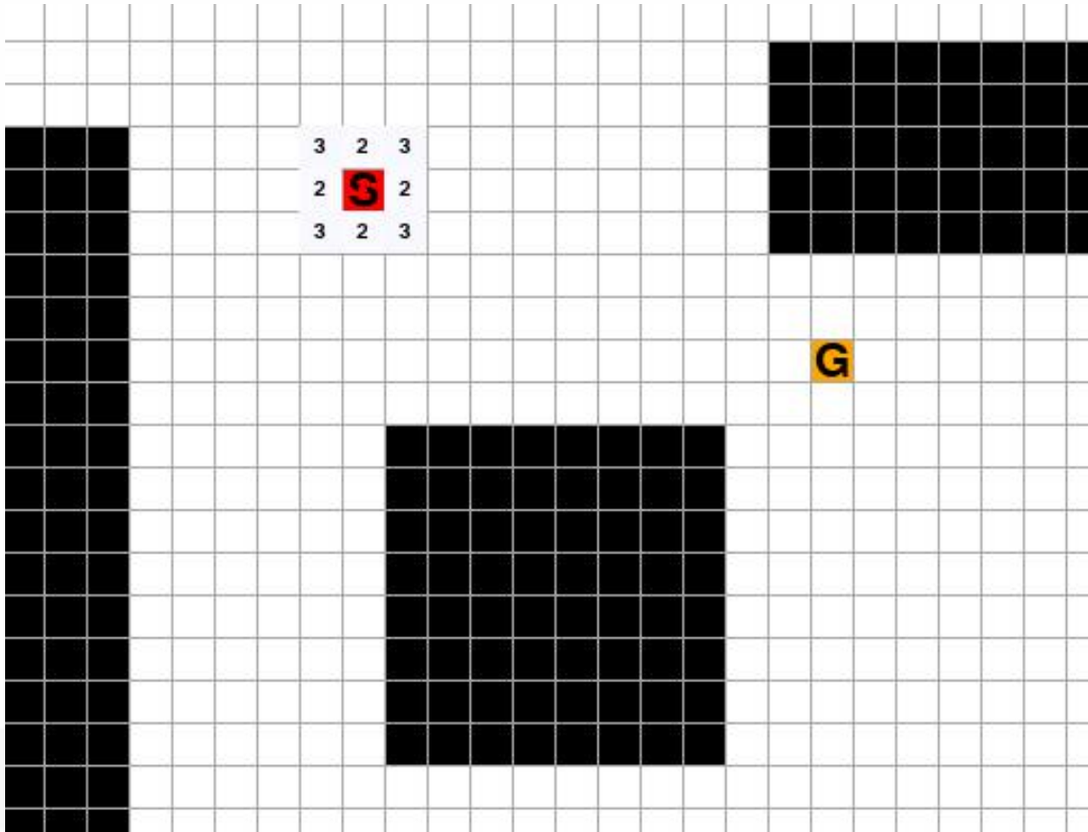
H代表从当前节点到终点**最小**估算路程，
使用曼哈顿距离、或欧几里得距离

A*算法过程讲解：



核心估算函数：F* = G + H*

情景假设：从起点开始
1.每次走一格，不能跨格
2.共有8个相邻栅格可以走
3.若走（上/下/左/右）计路程为：2
4.若走（左上/左下/右上/右下）计路程为：3

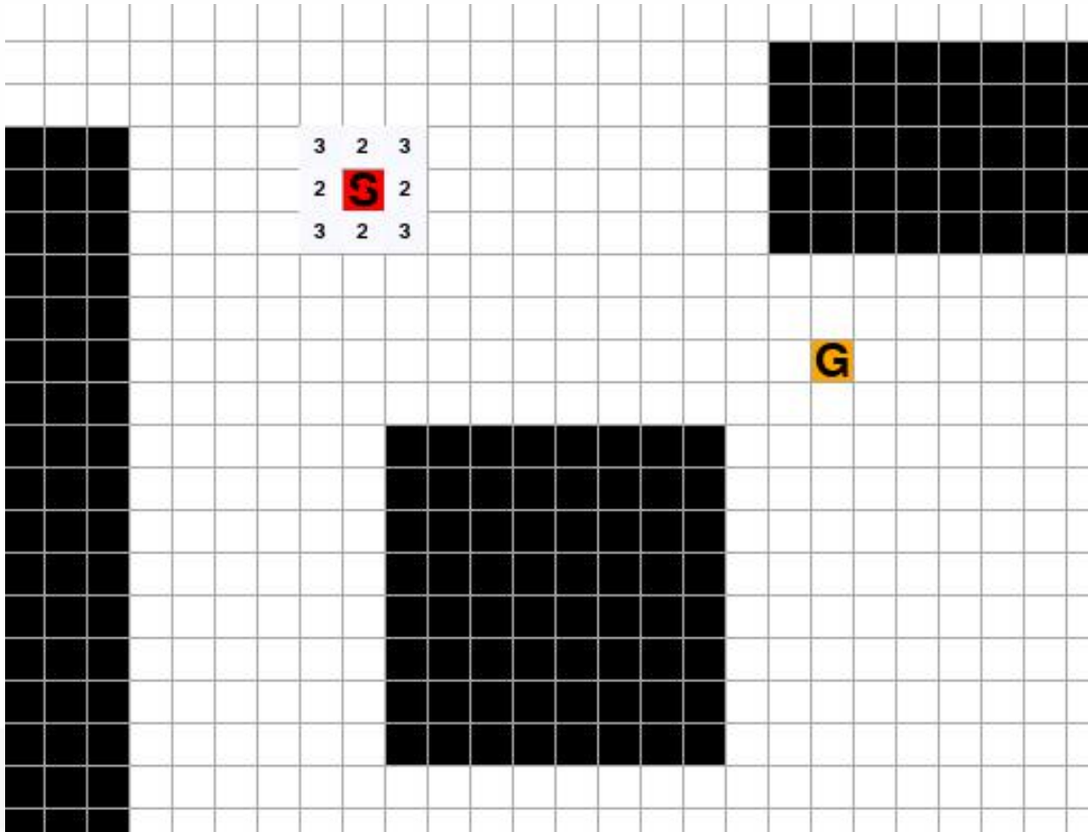


A*算法过程讲解：

A	B	C
H	S	D
G	F	E

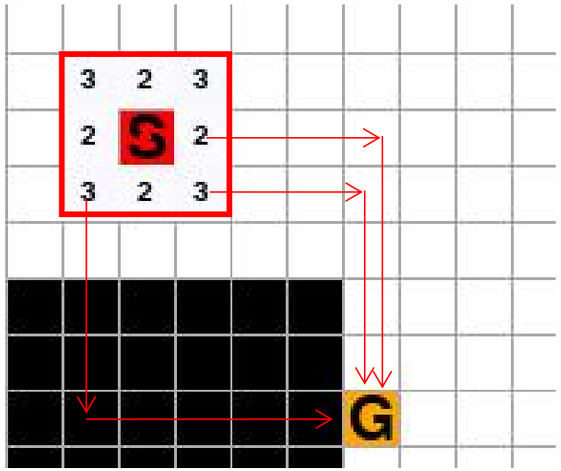
5.设置两个列表（或者数组）
openlist=[ ]　　——存放待确定路径的点
closedlist=[ ]　　——存放已确定路径的点
6.每个点都记录自己的父节点，下标表示
7.为了方便讲解
　　起点称S点
　　相邻8个点以A-H称呼
8.已放进closedlist的点不走，因已确定路径

9.算H*估算距离时，忽略障碍物



A*算法过程讲解：

①：起点为S点，计算周围点，父节点为S



A	B	C
H	S	D
G	F	E

$$F(D_S) = G(2) + H(3*2+5*2) = 18$$

$$F(E_S) = G(3) + H(3*2+4*2) = 17$$

$$F(G_S) = G(3) + H(4*2+5*2) = 21$$

.....

$$\text{openlist} = \begin{bmatrix} E_S = (3,14,17) \\ D_S = (2,16,18) \\ F_S = (2,16,18) \\ G_S = (3,18,21) \\ C_S = (3,18,21) \\ B_S = (2,20,22) \\ H_S = (2,20,22) \\ A_S = (3,22,25) \end{bmatrix} \quad \text{closedlist} = [S = (0,18,18)]$$

F* = G + H*

$$d(a,b) = |a_x - b_x| + |a_y - b_y|$$

A*算法过程讲解：

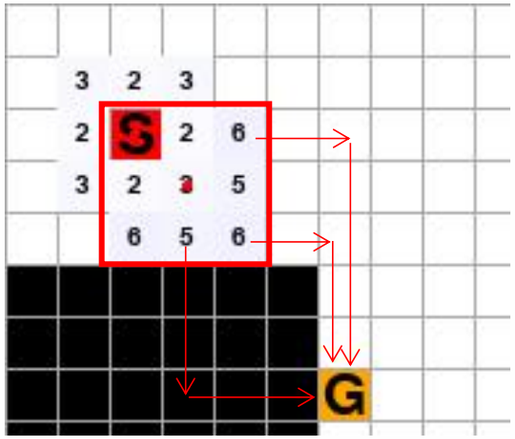
②：取F最小的点作第二步中心点E_S，重复步骤

$$F(C_{ES}) = G(6) + H(2*2+5*2) = 20$$

$$F(E_{ES}) = G(6) + H(2*2+3*2) = 16$$

$$F(F_{ES}) = G(5) + H(3*2+3*2) = 17$$

.....



A	B	C
H	S	D
G	F	E

从openlist取出，放入closedlist
计算中心点周围8个点，F = G + H
若点已存在openlist，取实际路程G小的方案

$$d(a,b) = |a_x - b_x| + |a_y - b_y|$$

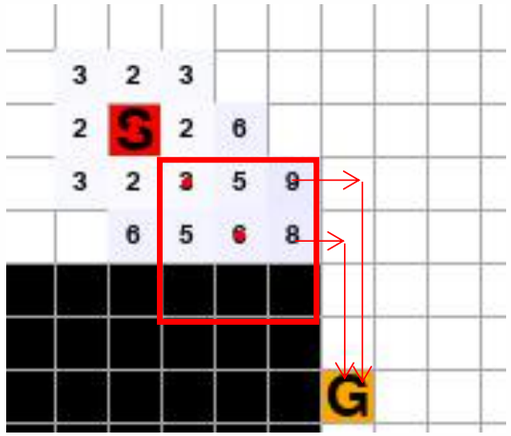
$$\text{openlist} = \begin{bmatrix} D_S = (2,16,18)...E_{ES} = (6,10,16) \\ F_S = (2,16,18)...D_{ES} = (5,12,17) \\ G_S = (3,18,21)...F_{ES} = (5,12,17) \\ C_S = (3,18,21)...C_{ES} = (6,14,20) \\ B_S = (2,20,22)...G_{ES} = (6,14,20) \\ H_S = (2,20,22) \\ A_S = (3,22,25) \end{bmatrix} \quad \text{closedlist} = \begin{bmatrix} S = (0,18,18) \\ E_S = (3,14,17) \end{bmatrix}$$

A*算法过程讲解：

③：取F最小的点作第三步中心点 E_{ES}，重复步骤

$$F(C_{EES}) = G(9) + H(1*2+4*2) = 19$$

$$F(D_{EES}) = G(8) + H(1*2+3*2) = 16$$



A	B	C
H	S	D
G	F	E

$$\text{closedlist} = \begin{bmatrix} S = (0,18,18) \\ E_S = (3,14,17) \\ E_{ES} = (6,10,16) \end{bmatrix}$$

从openlist取出，放入closedlist
计算中心点周围8个点，F = G + H
若点已存在openlist，取实际路程G小的方案

$$\text{openlist} = \begin{bmatrix} D_S = (2,16,18)...D_{ES} = (5,12,17)...D_{EES} = (8,8,16) \\ F_S = (2,16,18)...F_{ES} = (5,12,17)...C_{EES} = (9,10,19) \\ G_S = (3,18,21)...C_{ES} = (6,14,20) \\ C_S = (3,18,21)...G_{ES} = (6,14,20) \\ B_S = (2,20,22) \\ H_S = (2,20,22) \\ A_S = (3,22,25) \end{bmatrix}$$

$$d(a,b) = |a_x - b_x| + |a_y - b_y|$$

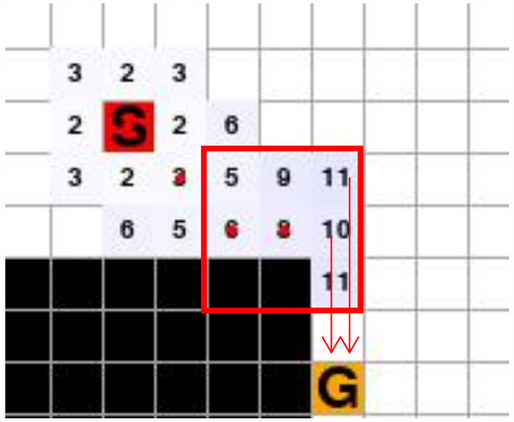
A*算法过程讲解：

④：取F最小的点作第四步中心点 D_{EES}，重复步骤

$$F(C_{DEES}) = G(11) + H(4*2) = 20$$

$$F(D_{DEES}) = G(10) + H(3*2) = 16$$

$$F(E_{DEES}) = G(11) + H(2*2) = 15$$



A	B	C
H	S	D
G	F	E

$$\text{closedlist} = \begin{bmatrix} S = (0,18,18)...D_{EES} = (8,8,16) \\ E_S = (3,14,17) \\ E_{ES} = (6,10,16) \end{bmatrix}$$

从openlist取出，放入closedlist
计算中心点周围8个点，F = G + H
若点已存在openlist，取实际路程G小的方案

$$\text{openlist} = \begin{bmatrix} D_S = (2,16,18)...D_{ES} = (5,12,17)...C_{EES} = (9,10,19) \\ F_S = (2,16,18)...F_{ES} = (5,12,17)...E_{DEES} = (11,4,15) \\ G_S = (3,18,21)...C_{ES} = (6,14,20)...D_{DEES} = (10,6,16) \\ C_S = (3,18,21)...G_{ES} = (6,14,20)...C_{DEES} = (11,8,20) \\ B_S = (2,20,22) \\ H_S = (2,20,22) \\ A_S = (3,22,25) \end{bmatrix}$$

$$d(a,b) = |a_x - b_x| + |a_y - b_y|$$

A*算法过程讲解：

⑤：取F最小的点作第五步中心点E_{DEES}，重复步骤

A	B	C
H	S	D
G	F	E

从openlist取出，放入closedlist
计算中心点周围8个点，F = G + H
若点已存在openlist，取实际路程G小的方案

$$d(a,b)=|a_x-b_x|+|a_y-b_y|$$

$$F(F_{EDEES})=G(13)+H(1*2)=15$$

$$F(C_{EDEES})=G(14)+H(1*2+3*2)=22$$

$$F(D_{EDEES})=G(13)+H(1*2+2*2)=19$$

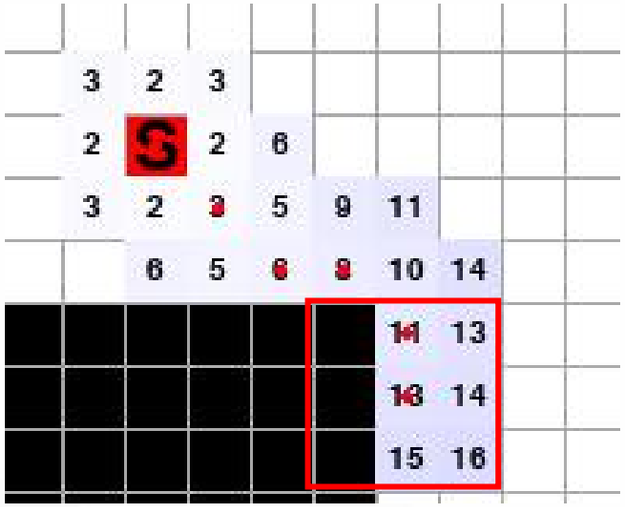
.....

$$\text{closedlist}=\begin{bmatrix} S=(0,18,18)...D_{EES}=(8,8,16) \\ E_S=(3,14,17)...E_{DEES}=(11,4,15) \\ E_{ES}=(6,10,16) \end{bmatrix}$$

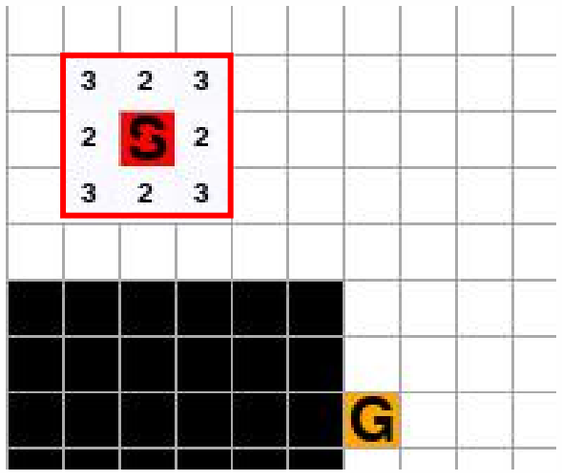
$$\text{openlist}=\begin{bmatrix} .....C_{EES}=(9,10,19)...F_{EDEES}=(13,2,15) \\ .....D_{DEES}=(10,6,16)...E_{EDEES}=(14,4,18) \\ .....C_{DEES}=(11,8,20)...D_{EDEES}=(13,6,19) \\ .....C_{EDEES}=(14,8,22) \\ ..... \\ ..... \end{bmatrix}$$

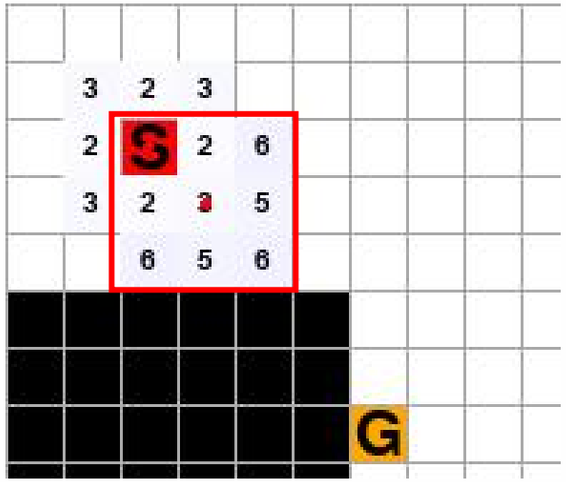
A*算法过程讲解：

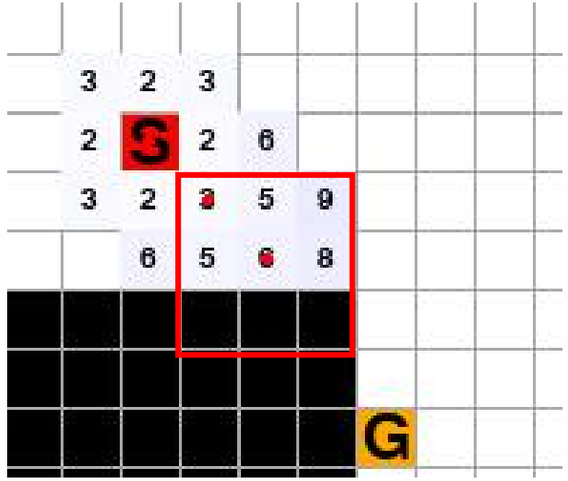
⑥：取F最小的点作第六步中心点 F_{EDEES}，重复步骤

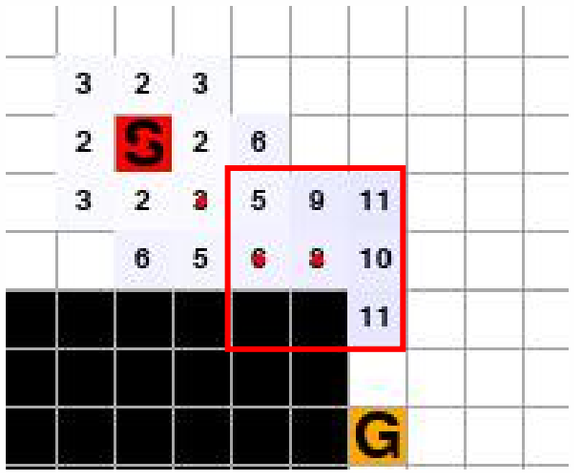


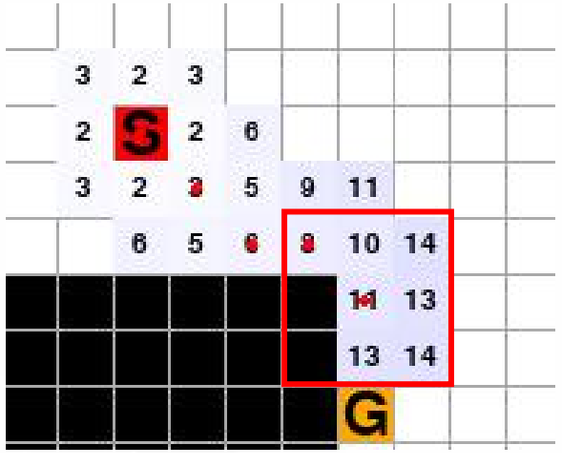
第七步，找到目标点，通过父节点反推，提取路径

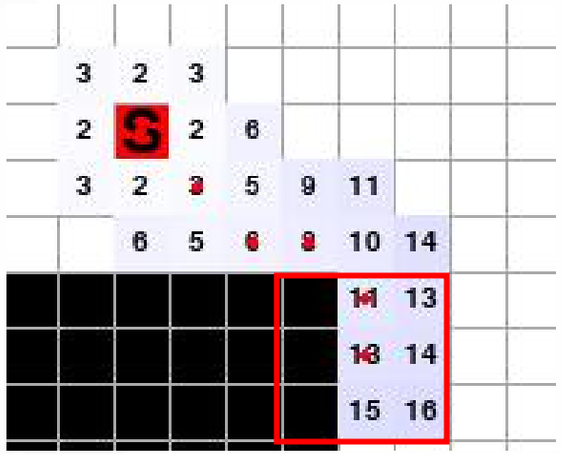

3 2 3
2 S 2
3 2 3
G


3 2 3
2 S 2 6
3 2 3 5
6 5 6
G


3 2 3
2 S 2 6
3 2 3 5 9
6 5 6 8
G


3 2 3
2 S 2 6
3 2 3 5 9 11
6 5 6 8 10
11
G


3 2 3
2 S 2 6
3 2 3 5 9 11
6 5 6 8 10 14
14 13
13 14
G


3 2 3
2 S 2 6
3 2 3 5 9 11
6 5 6 8 10 14
14 13
13 14
15 16


3 2 3
2 S 2 6
3 2 3 5 9 11
6 5 6 8 10 14
11 13
13 14
G 16

伪代码：

```
获取起点坐标、目标点坐标，算出起点的F值(F=0+H)把起点放进openlist
while (openlist不为空列表)
{       取openlist中F值最小的点为中心点，从openlist剔除，放入closedlist
        if (中心点是目标点)
                {从目标点开始通过父节点反推，提取出路径，break跳出循环}
        else
                {遍历相邻的八个点
                        {if(点已存在closedlist中  or  遇到障碍物 )
                                {跳过，不计算这个点}
                        else
                                {累计实际路程G，估算H，F=G+H
                                if(点已存在openlist中)
                                        if(实际路程G<原来记录的G)
                                                {替换数据}
                                else {加入openlist}
                        }
                openlist从小到大排序
        }
}
```

WHEELTEC
轮 趣 科 技

THANK YOU

感谢聆听