

A*算法详细讲解以及实现

原创 ypw44 于 2020-05-24 16:08:12 发布 阅读量1.7w 收藏 97 点赞数 23

分类专栏: 人工智能

人工智能 专栏收录该内容

2 订阅 13 篇文章

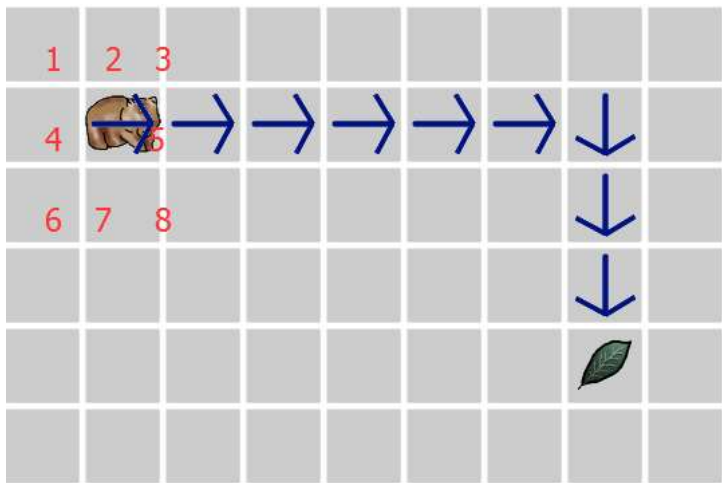
A*简介

A算法是启发式算法重要的一种，主要是用于在两点之间选择一个最优路径，而A的实现也是通过一个估值函数

F=G+H

- G表示该点到起始点位所需要的代价
- H表示该点到终点的曼哈顿距离。
- F就是G和H的总和，而最优路径也就是选择最小的F值，进行下一步移动（后会做详细介绍）

曼哈顿距离

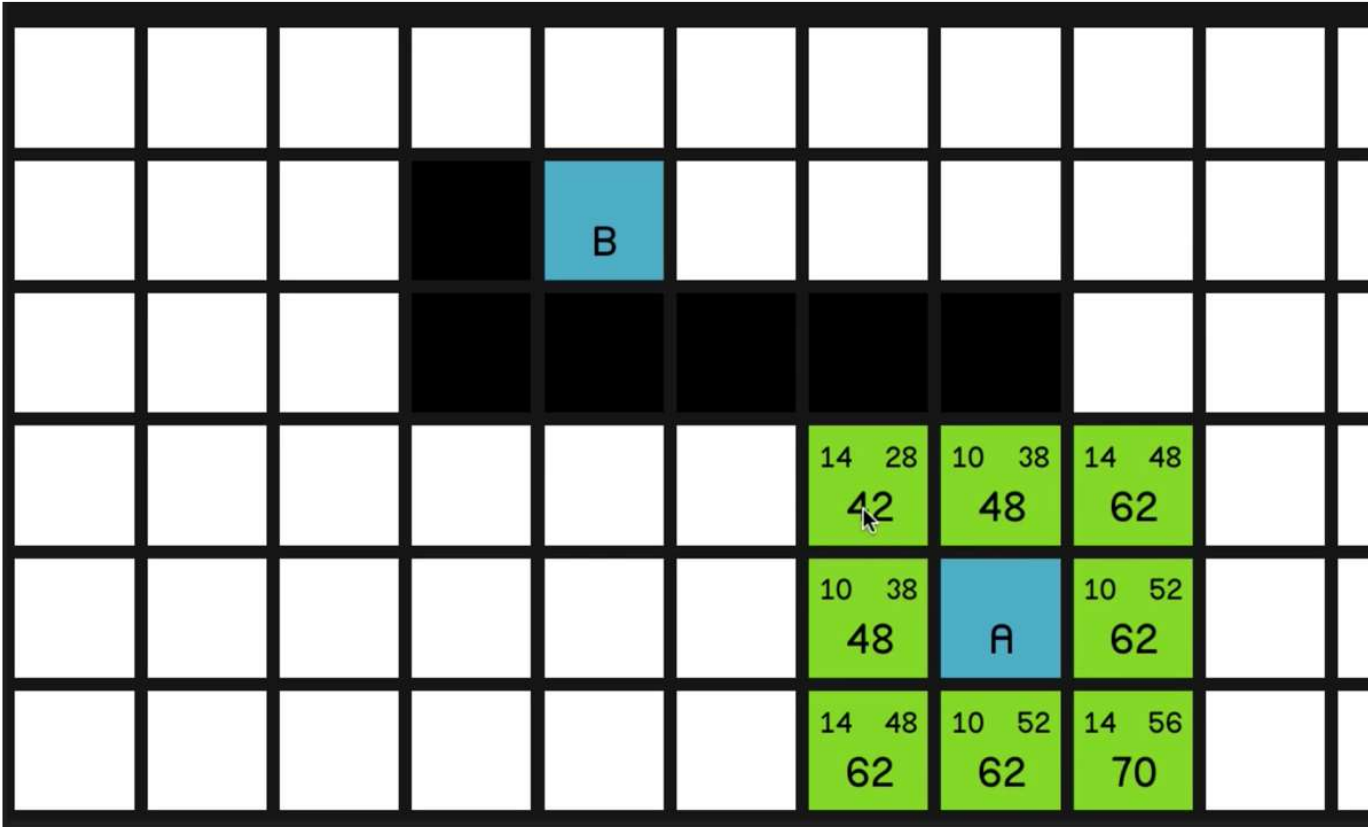


Paste_Image.png

上图中这个熊到树叶的曼哈顿距离就是蓝色线所表示的距离，这其中不考虑障碍物，假如上图每一个方格长度为1，那么此时的熊的曼哈顿距离就为9
起点 (X1,Y1) ,终点 (X2,Y2) , $H=|X2-X1|+|Y2-Y1|$

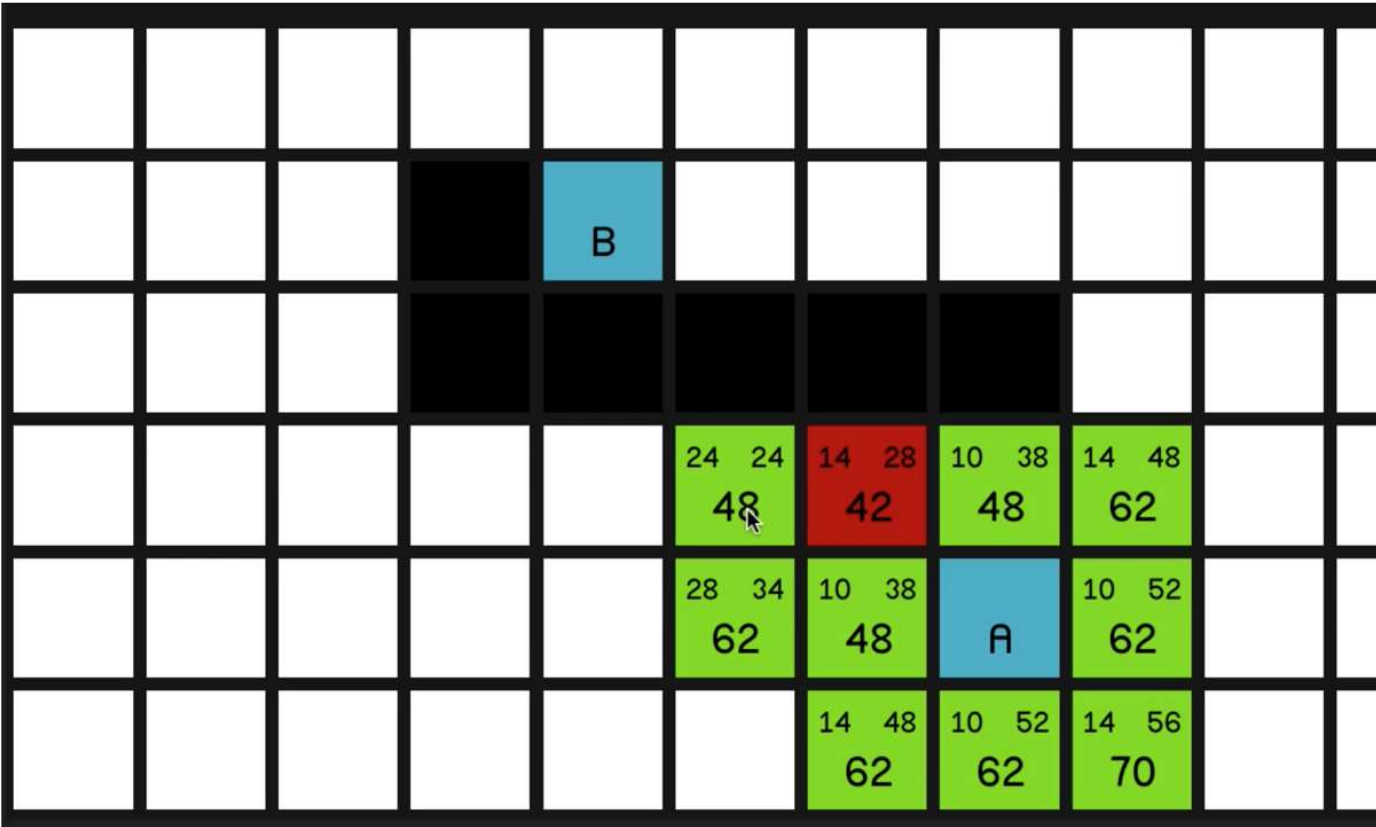
我们也可以通过几何坐标点来算出曼哈顿距离，还是以上图为例，左下角为 (0, 0) 点，熊的位置为 (1, 4) ，树叶的位置为 (7, 1) ，那么 $H=|7-1|$

A算法现在被广泛应用与电脑游戏中的路径规划问题。我们就以此为例来介绍A算法的具体实施步骤。如下图所示，其中A表示起点，B表示终点，黑色块表示障碍物。此外我们假设水平或垂直方向上相邻的两个方块之间距离是10，那么对角线方向上相邻的两个方块距离就约是14。

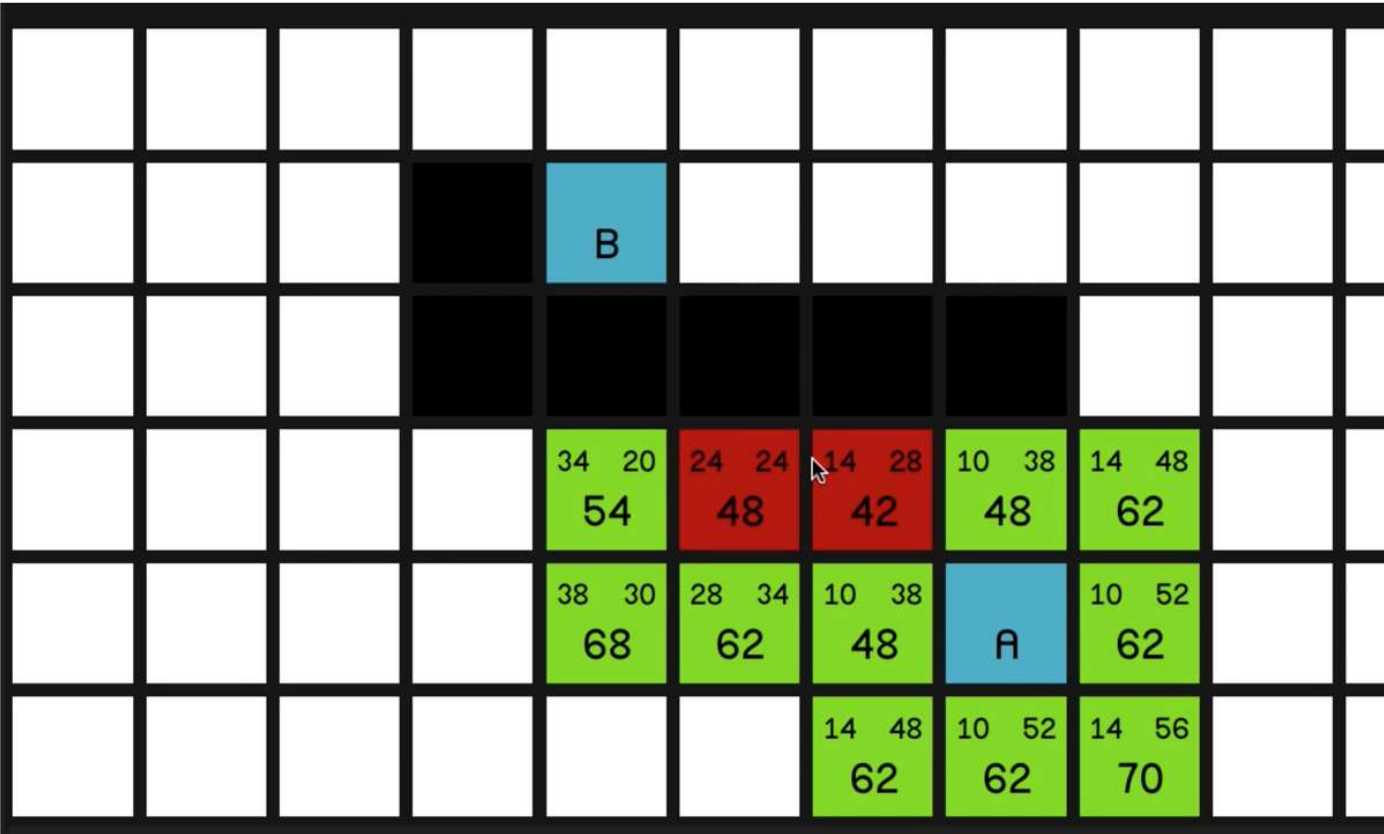


算法开始，我们首先搜索A相邻的所有可能的移动位置（对应于图中的绿色方块）。每个方块左上角的值G表示该点到A的距离，右上角值为H，注意该点到B的距离，所以这里的H就取其到B的距离。最后，还要计算一个F值， $F = H + G$ 。

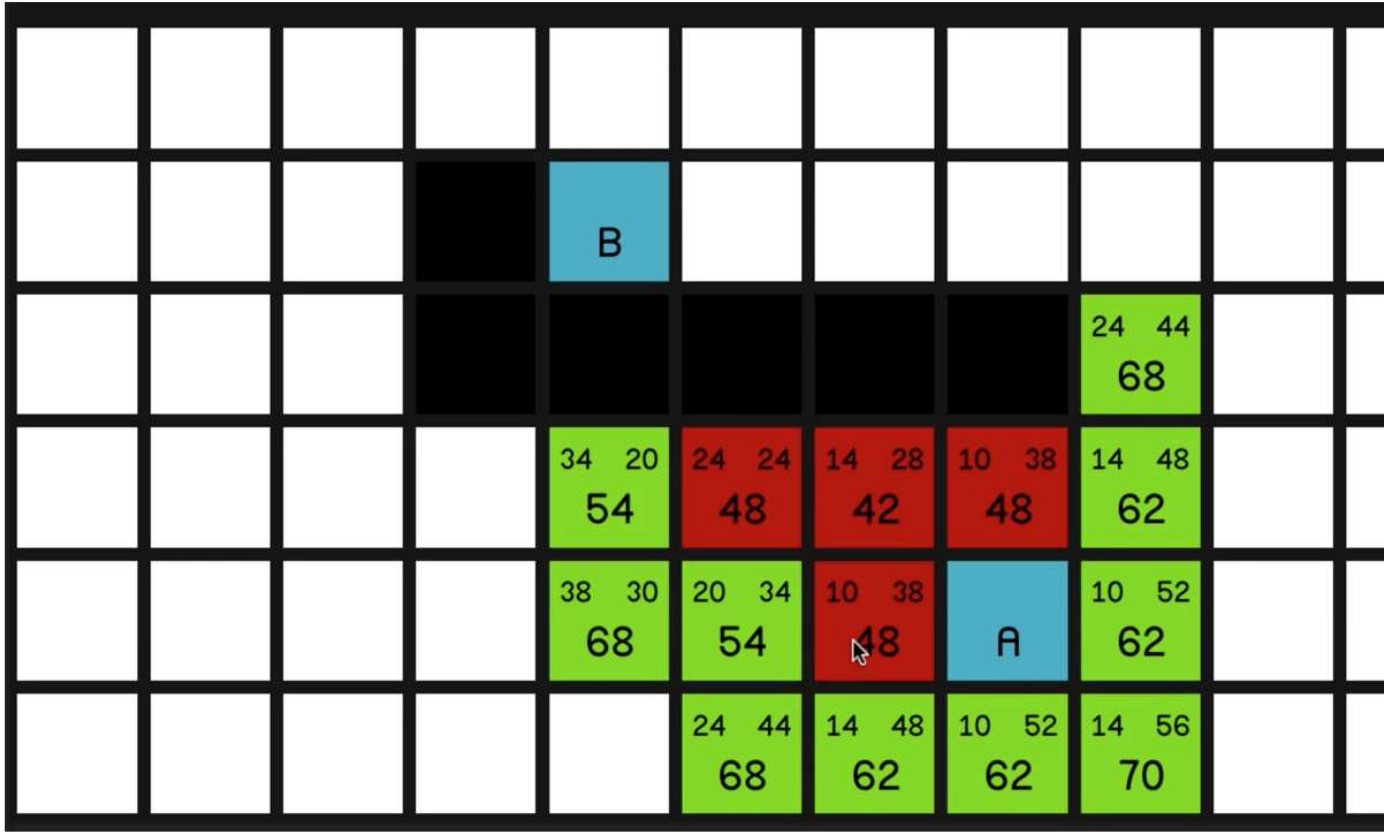
然后像Dijkstra算法一样，我们选一个F值最小的节点来做继续搜索。也就是上图中A的邻域中位于左上角的值（ $F = 42$ ）。然后更新该节点的领域值。



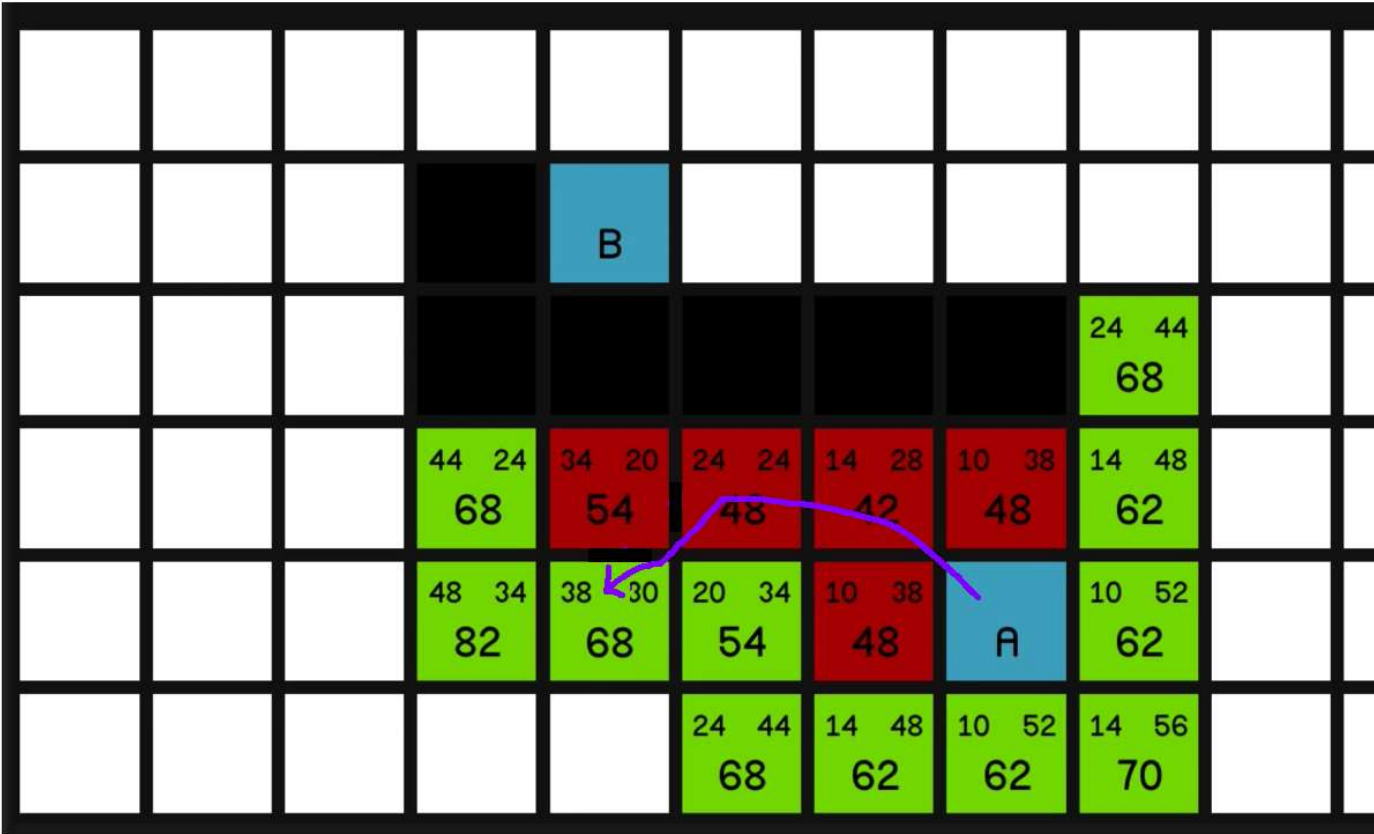
这时你会发现，出现了三个F值都等于48的节点。到底应该选择哪一个来继续接下来的搜索呢？这时需要考察它们中的那个H值最小，结果发现 $H = 24$ ，所以下面就要从该点出发继续搜索。于是更新该节点的邻域方块中的值。



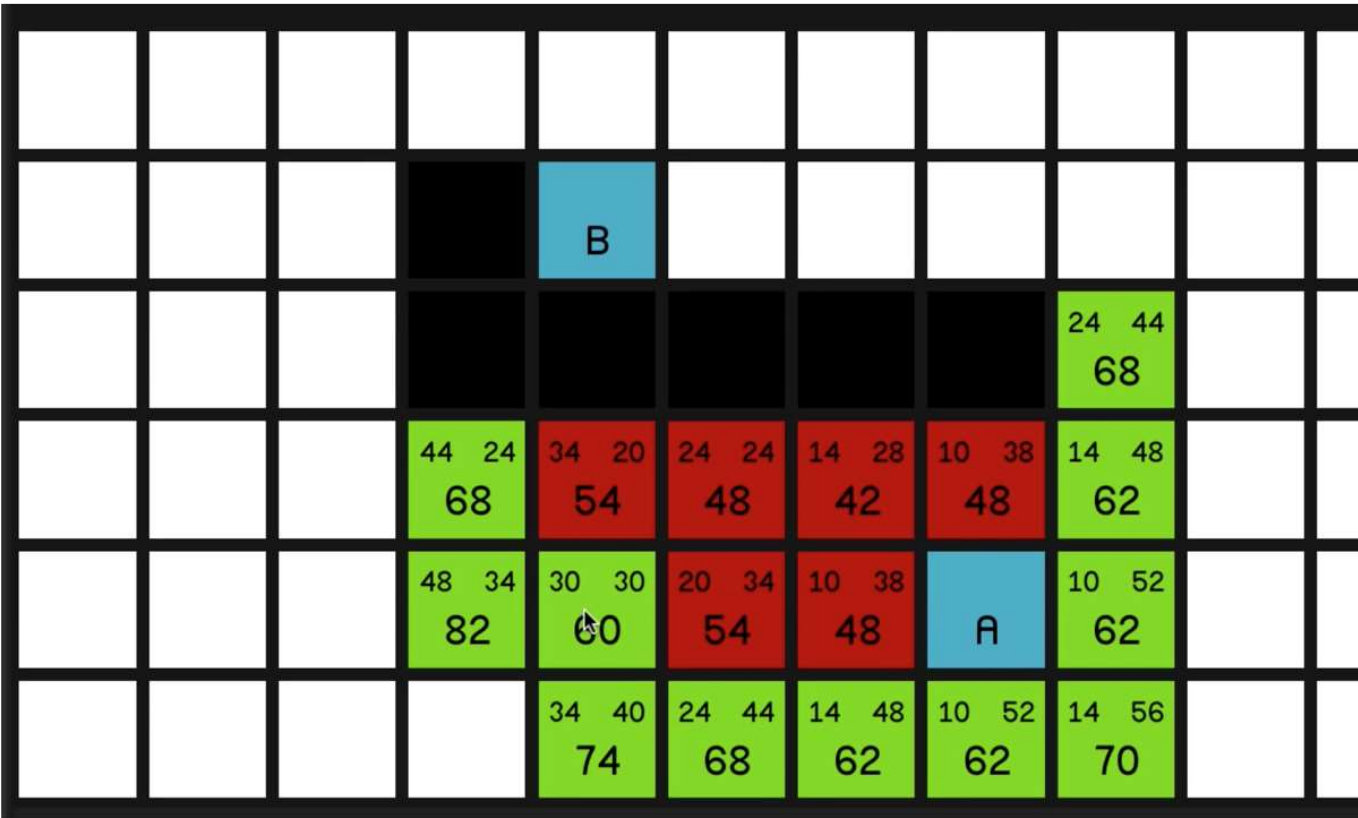
这个时候再找出全局F值最小的点，结果发现有两个为48（而且它们的H值也相当），于是随机选取一个作为新的出发点并更新其邻域值（例如选择右块），然后在从全局选取F最小的更新其邻域值，于是有：



此时全局F最小的值为54，而且F=54的节点有两个，所以我们还是选择其中H值最小的来做更新。于是更新该节点邻域方块中的值。这里你需要注意的是，F=54的红色节点下方邻域（F=68）的方块中，G=38。但是，从A到该节点的最短路径应该是30。这是因为目前程序所选择的路径是下图中紫色标出来的路径，其G的增长序列是14→24→38。



不过不要紧，只要继续执行算法，更新全局F值为最小节点（F=54）的方块，上面的G值就会给更新为正确的值了。



此时，全局F值最小的方块中F=60，所以更新该节点邻域方块中的值。

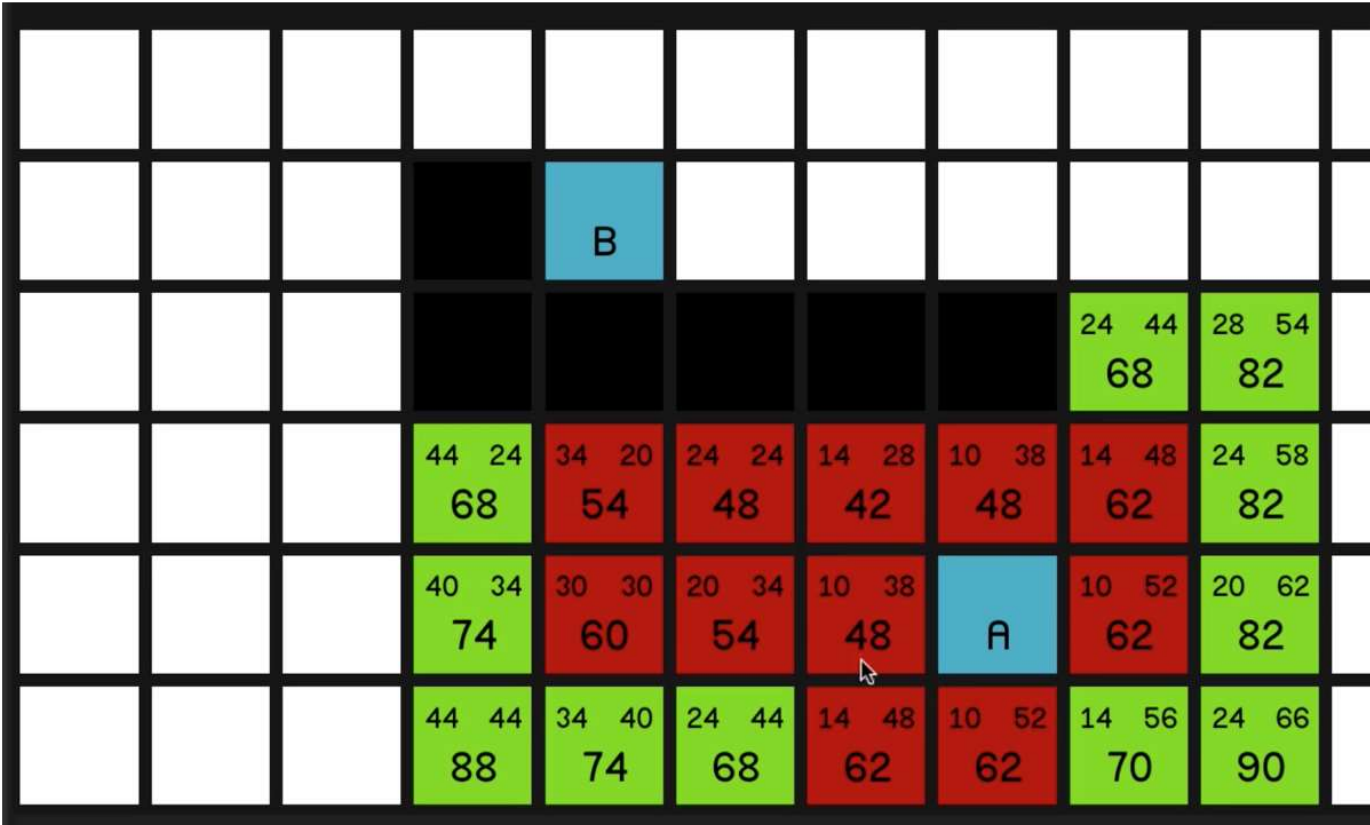


ypw44

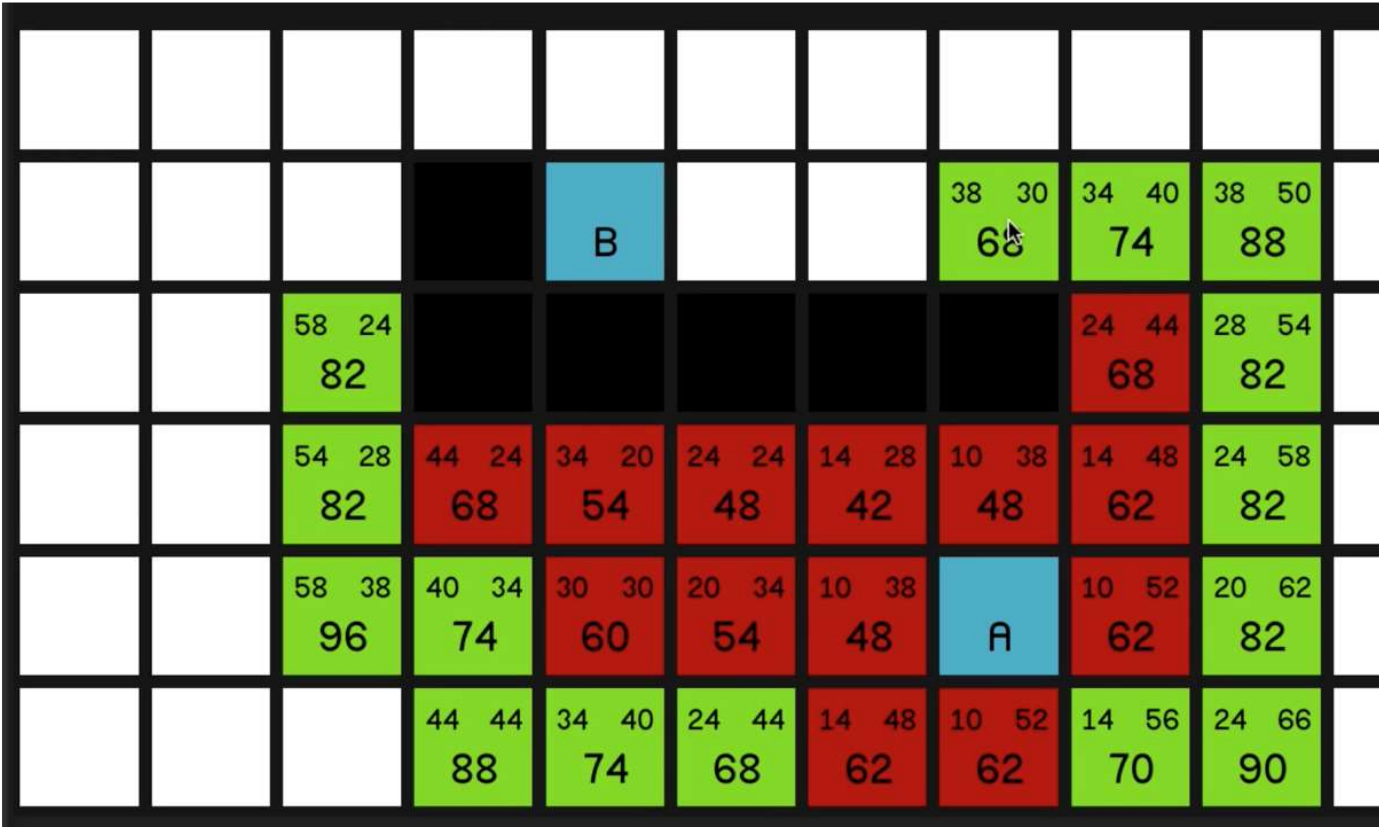
关注



23



现在全局F值最小的有两个，都为68，此时先更新H最小的。这是其实程序已经发现左侧F=68的节点并不能引导一条更短的路径。于是接下来就要转向



最终反复执行上述过程，你就会得到如下图中蓝色方块所示的一条最短路径。