AStar 算法介绍的复制

目录

概述

AStar (A*)算法是一种启发式搜索算法，在Dijkstra的基础上，增加了一个目的导向函数（启发函数），精简下来的函数f(n) = g(n) + h(n)。

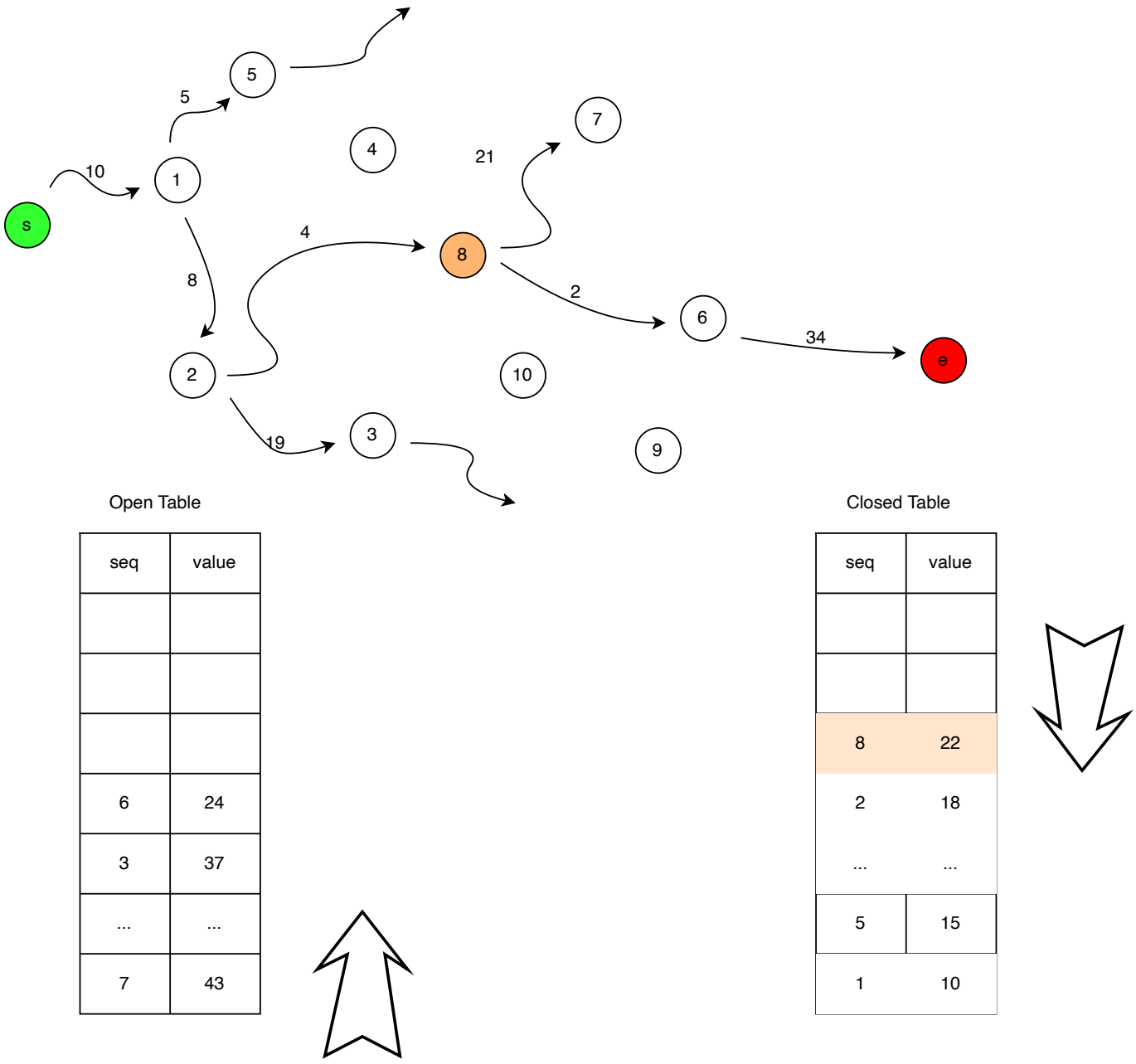
- f(n) 是从初始点经由节点n到目标点的估价函数。
- g(n) 是在状态空间中从初始节点到n节点的实际代价。
- h(n) 是从n到目标节点最佳路径的估计代价。

Dijkstra

Dijkstra 算法是最基础的搜索方法之一，是一种贪婪模式算法，函数只考虑当前从起点到当前节点的实际代价{ f(n) = g(n) }。

探索方式很简单，就是计算每一个后继节点的代价，然后就比较节点的代价，先探索代价小的，直到探索到终点。

要实现Dijkstra我们一般使用优先队列（priority queue），简单来说就是一个open优先队列和一个closed集合的组合，open优先队列用来储存还没有遍历将要遍历的节点，而closed集用来储存已经被遍历过的节点，**直到终点在closed集合中出现，结束探索**。这种实现方式也会运用到其他算法上，只是给open优先队列中的节点排序不同。

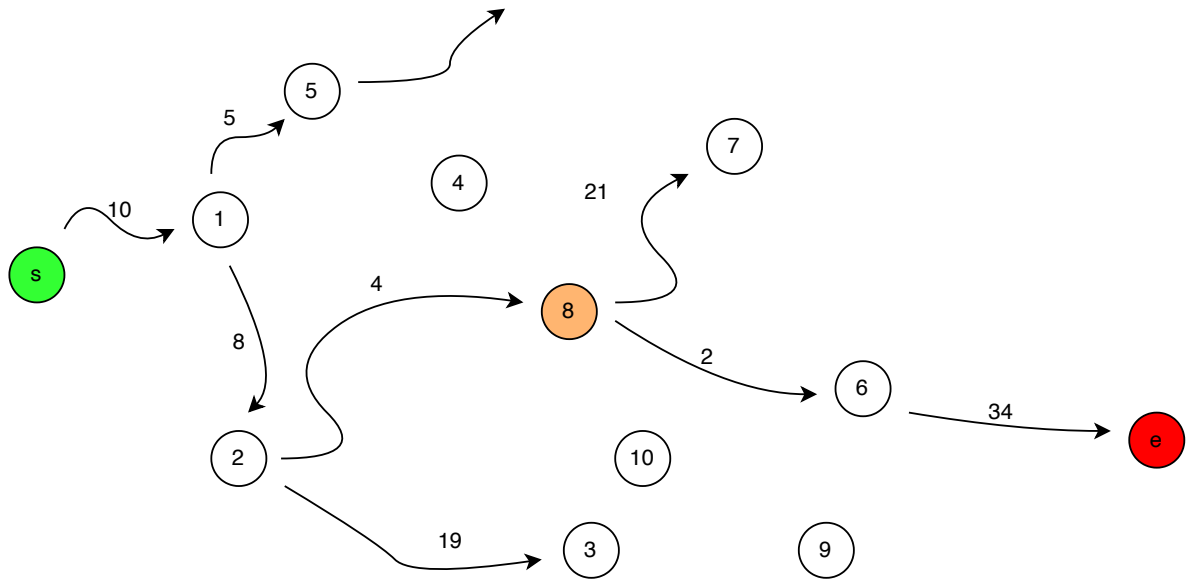


BFS

最佳优先搜索（Best first search），这个算法的核心思想是：**每一步会选择离目标点最近的结点**。Best-First 算法是一种贪心算法，一般通过定义一个启发式函数来引导着向离目标更近的方向前进，常见的启发式函数为欧氏距离(Euclidean Distance)或者曼哈顿距离(Manhattan Distance)。[不是广度优先搜索算法（ Breadth First Search , BFS)]

BFS算法用启发估价函数对将要被遍历到的点进行估价，然后选择代价小的进行遍历，直到找到目标节点或者遍历完所有点，算法结束。

但是有可能找到的路径不是最优路径。



cost(节点->End)

seq	ToEnd
1	55
2	48
3	7
4	29
5	60
6	5
7	18
8	20
9	9
10	18
...	...

Open Table

seq	value
6	5
7	18
...	...
5	60

Closed Table

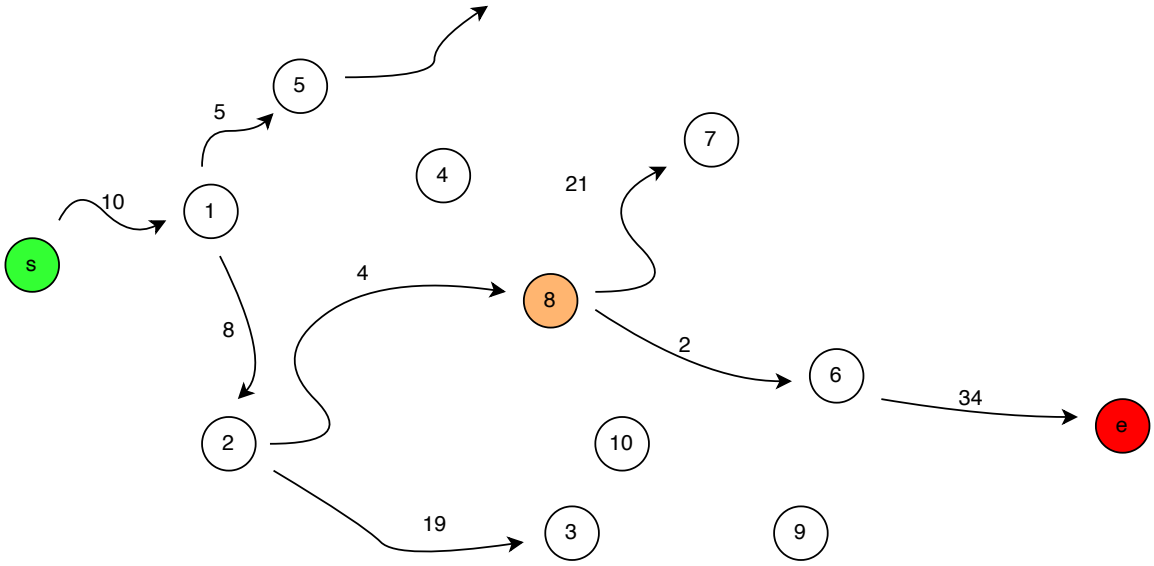
seq	value
8	20
3	37
2	18
...	...
1	10

Astar 算法

A算法也是一种启发式搜索算法，但相比之下，它综合了节点的实际代价和启发式评估函数的值来选择节点进行扩展。A算法使用一个估计函数f(n)，它将节点的实际代价（从起始节点到达当前节点的代价，通常称为g(n)）和节点到目标节点的估计代价（启发式评估函数，通常称为h(n)）结合起来，计算节点的综合评估值f(n) = g(n) + h(n)。然后，A算法选择具有最低综合评估值的节点进

行扩展。通过综合考虑实际代价和启发式评估函数的值，A算法能够更加智能地指导搜索，以找到一条效率更高的路径。

条件下可以找到最优路径，条件就在启发函数上。 f(n) = g(n) + h(n) ， 如果0<= h(n) <= 真实的n到达终点的代价，是可以获取最优路线的，如果h(n) > 真实代价，则会很快探索到终点，得到一条非最优路径。



cost(节点->End)

seq	ToEnd
1	55
2	48
3	7
4	29
5	60
6	5
7	18
8	20
9	9
10	18
...	...

Open Table

seq	value
6	24+5
7	43+18
...	...
5	15+60

Closed Table

seq	value	
8	22	22+20
3	37	37+7
2	18	18+48
...	...	
1	10	10+55

名词解释

- 启发式算法

启发式算法（heuristic algorithm)是相对于最优化算法提出的。一个问题的最优算法求得该问题每个实例的最优解。启发式算法可以这样定义：一个基于直观或经验构造的算法，在可接受的花费（指计算时间和空间）下给出待解决组合优化问题每一个实例的一个可行解，该可行解与最优解的偏离程度一般不能被预计。

参考文献

自动驾驶路径规划——A*（Astar）算法

算法仿真环境

QA