

Introduction to Aerial Robotics

L4

- Search-based Method
 - 对于每一个搜索问题都会有对应的状态空间图（state space graph）
 - 在图中的点与点之间的联系是由边呈现的
- From Graph to Search Tree
 - 搜索图生成一个搜索树，回溯搜索树中的一个节点能够产生一条从开始状态到那个节点的路径
 - 由于太大或者不高效，通常不会构造全部的树
- Depth First Search（DFS）

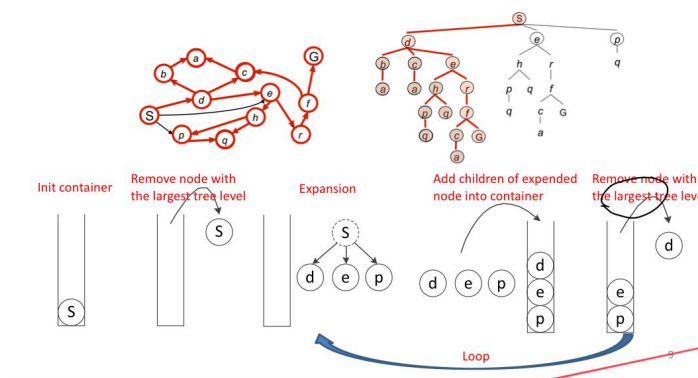


香港科技大学
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

电子及计算机工程系
DEPARTMENT OF ELECTRONIC & COMPUTER ENGINEERING

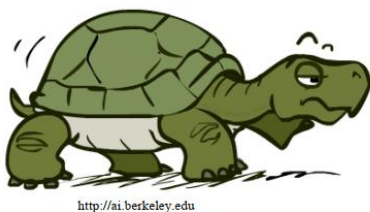
Depth First Search (DFS)

- Implementation: maintain a last in first out (LIFO) container (i.e. stack)



- 使用后进先出的策略（LIFO）
- Breadth First Search（BFS）
 - 使用先进先出的策略
- Dijkstra's algorithm
 - 使用最小积累成本 $g(n)$ 【the cheapest accumulated cost】
 - $g(n)$ ：当前从开始状态到节点 n 的积累成本的最好的估计；

- 更新从 n 到还没有扩展到 m 到积累成本 $g(m)$
- 一个节点确保从开始状态到当前节点为最小成本
- 其实可以看作是有成本的 BFS，因为他使用一种优先队列（priority queue）去储存全部已经拓展的节点。
- 由于 dijkstra 算法并不是一种目标导向型的搜索算法，所以计算量还是很大。
- Search Heuristics（启发性算法）
 - greedy best search
 - 只有启发但是没有积累成本
 - A* Search: 结合了 Dijkstra 和 Heuristic
 - Heuristic: $h(n)$ 从 n 节点到目标节点的估计最小成本（estimated least cost）
 - 所以 $f(n) = g(n) + h(n)$
 - 预估成本 $h(n)$ 可以选用欧几里得距离或者曼哈顿距离，曼哈顿距离不用开方，计算速度快。
 - 由于有 $h(n)$ 的存在，所以 A* 算法能够导引到目标的位置。
-
- 相关网址 <https://www.redblobgames.com/pathfinding/a-star/introduction.html>



UCS



Greedy



A*

<https://blog.csdn.net/fuozjw>