

19024075 陶盛皿 Week4 Assignment

课后习题3.6：树搜索方法和图搜索方法

Name	Tree Search	Graph Search
算法返回值：return	A solution or failure	
初始化：init	The initial state of problem	
有无已扩展结点集合搜索集（closed集）	无	有，记录已扩展结点
循环：loop 终止条件	新开辟空间frontier（stack, deque, hash, etc.）为空	
循环：loop过程	node = frontier.pop(), return对应的solution或者action	
explored set	无	explored_set.append(node)
expand node	展开node，将node.child加入frontier	展开node，判断node.child是否在explored_set，如果不是加入frontier
有无有环路径（冗余）	有	无，将状态空间划分成已探索区域和未探索趋于

课后习题3.8：无信息搜索和有信息搜索

Name	Uninformed Search	Informed Search
定义	除了问题定义提供的状态信息外没有任何附加信息	除了问题定义提供的状态外还知道非目标和其他状态的信息
别名	盲目搜索	启发式搜索
完备性	有限状态空间完备	A*完备
复杂度	时间、空间复杂度受限于状态空间规模	时间、空间复杂度受限于状态空间规模或者cut-off或者有限内存
算法	BFS; UCS; DFS; DLS; IDS; Bi-S	最佳优先搜索；贪婪最佳优先搜索；A*；RBFS; SMA*
代价函数	UCS有g(n)	基于f(n)评价被选择的结点
启发函数	无	h(n)最小代价路径的代价估计值

课后习题3.9：有信息评价函数 $f(n) = g(n) + h(n)$

评价函数： $f(n) = g(n) + h(n)$ = 开始结点到n的路径代价 + n到目标结点的最小代价路径的估计值 = 经过n的最小代价估计

简述无信息搜索

1. 无信息搜索（盲目搜索）策略以节点扩展的次序来分类的（宽度优先，一致代价，深度优先，深度受限，迭代加深，双向搜索）。
2. 性能评价：
 1. 完备性：是否总能找到一个存在的解。
 2. 时间复杂度
 3. 空间复杂性
 4. 最优性：是否找到最优解。

简述宽度优先搜索

- **BFS思路**：先扩展root，再扩展root.child，然后在扩展root.child.child，如此继续。
- **队列**：存放待展开结点，FIFO，先扩展浅结点再扩展深结点。
- **终止条件**：找到解就终止。
- **优点**：完备性，如果深度d和分支因子b都有限，则存在目标结点，但不一定最优；如果path_cost非递减，则算法代价有最优解。

- **缺点：**在深度d比较大的时候，时间和空间都会指数爆炸。
- **空间复杂度：** $O(b^d)$
- **时间复杂度：** $O(b^d)$

简述深度优先搜索

- **DFS思路：**调用递归函数，扩展边缘结点集中最深的结点，然后向上回溯稍浅的未扩展结点。
- **栈：**LIFO
- **终止条件：**
 - 迭代加深，循环到设置的深度终止限制cut-off = l;
 - 递归遍历后无目标结点failure;
 - (特殊情况) 根节点None。
- **优缺点：**
 - 如果有限状态空间，完备;
 - 如果树搜索，不完备;
 - 如果无限空间，会冗余，可能死循环。
 - 如果深度受限，
 - $l < d$, 不完备。
 - $l > d$, 不是最优。
- **空间复杂度：** $O(bm)$, m是任一结点深度, $m = l = \text{cut-off}$ 或者趋于无穷。
- **时间复杂度：** $O(b^m)$

简述一致代价搜索

- **UCS思路：**扩展路径消耗g(n)最小的结点，将边缘结点按g值排序的队列；在结点被选择扩展是进行测试是否是最优路径上，并且path_cost非递减（每步代价>=某个正常数是完备，=0会死循环），故按结点的最优路径顺序扩展结点，被选择扩展的结点是最优解。
- **队列：**按照path_cost排序；开辟explored set。
- **终止条件：**选择代价最小。
- **优点：**完备性；代价最优。
- **缺点：**时间空间复杂度比BFS大。
- **空间复杂度：** $O(b^{1+\lceil C^*/\epsilon \rceil})$
- **时间复杂度：** $O(b^{1+\lceil C^*/\epsilon \rceil})$

简述有信息搜索

1. 启发式搜索：除了问题定义外有的启发信息来引导搜索，达到减少搜索范围、降低问题复杂度的目的。(最佳优先搜索，贪婪最佳优先搜索，A*搜索，...)
2. 有评价函数： $f(n) = g(n) + h(n)$ 或者 $f(n) = h(n)$ ，基于评估函数f(n)选择扩展结点，通过f(n)对优先级队列排队。

简述贪婪搜索

- **思路**：每一步扩展 $h(n)$ 最小的结点。
 - **评价函数**： $h(n)$ ，单步路径代价函数。
 - **优点**：效率最高。
 - **缺点**：不最优；不完备。
 - **复杂度**：时间空间复杂度都是 $O(b^m)$ ，依赖于状态空间和启发函数。
-

简述A star 搜索

- **评价函数**： $f(n) = g(n) + h(n)$ = 开始结点到 n 的路径代价 + n 到目标结点的最小代价路径的估计值 = 经过 n 的最小代价估计
 - **队列**：按照 $path_cost$ 排序；开辟explored set。
 - **终止条件**：选择代价最小。
 - **优点**：完备；最优。
 - **缺点**：时间空间复杂度比BFS大。
 - **最优性条件**：
 - 可采纳启发式： $f(n) = g(n) + h(n) \leq g(n) + h(n)$ 实际代价
 - 一致性： $h(n) \leq$ 从 n 到 n' 的单步代价 + $h_hat(n')$
 - **复杂度**：外循环中每次从open中取出点，共取 n 次；内循环：遍历它的邻接点 $n(E)$ ，并将这些邻接点放入open中，对open进行排序，open表大小是 $O(n)$ 量级的，若用快排就是 $O(n \log n)$ ，内循环总的复杂度为 $O(n \log n + E) = O(n \log n)$ ，总复杂度为 $O(n^2 \log n)$ 。
-