创建一个小根堆 heap (优先队列，按照 node.cost排序)

令 父节点 = nil， action = nil

创建一个 Node(父节点，action，worldStatus，agentGoal，cost) 并加入到 heap

遍历小根堆，获取堆顶节点 node

获取所有可执行的 ActionList

按照 cost 从小到大排序

按照 node.parentNode 一直遍历到跟节点，并按顺序保存，根节点就是第一个可执行的 action 路径起点，倒叙取出就是可执行 action 路径

如果 node.action 不为空

并且 action 的先决条件都包含在 node.worldStatus

Y

N

遍历ActionList 检测每一个 action

判断action是否可用

N

Y

从 node.worldStatus 复制一份 新环境变量：newWorldStatus  
从 node.agentGoal 复制一份 新的目标：newAgentGoal

action.preCondition 所有值 不与 newAgentGoal 冲突

1.将 action 的执行效果(action.effect)，添加到 新的环境变量(newWorldStatus)中

2.将 action 的执行效果(action.effect)，从 新Agent目标(newAgentGoal)中移除

3.将 action 的先决条件(action.preCondition) 添加到 newAgentGoal

4.创建一个节点 Node(node，action，newWorldStatus，newAgentGoal，node.cost + action.cost) 并加入到 heap

action.effect 至少一条满足 newAgentGoal

N

Y

N

Y