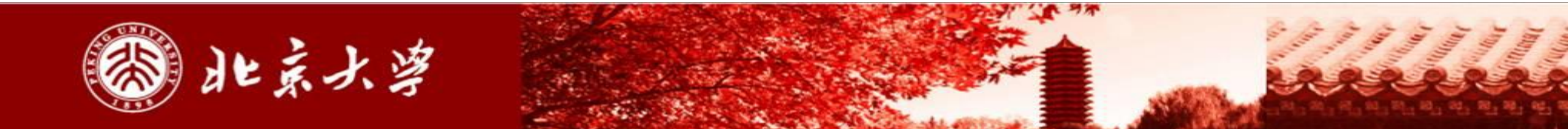
《物理与人工智能》

14. 搜索概览

授课教师：马滟青

2026/10/26（第七周）

鸣谢：基于计算机学院《人工智能引论》课程组幻灯片



目录

- 目标：开始尝试构建一个能解决问题的智能体
- 什么是搜索
- 如何定义搜索问题
- 常见的搜索策略

搜索与智能

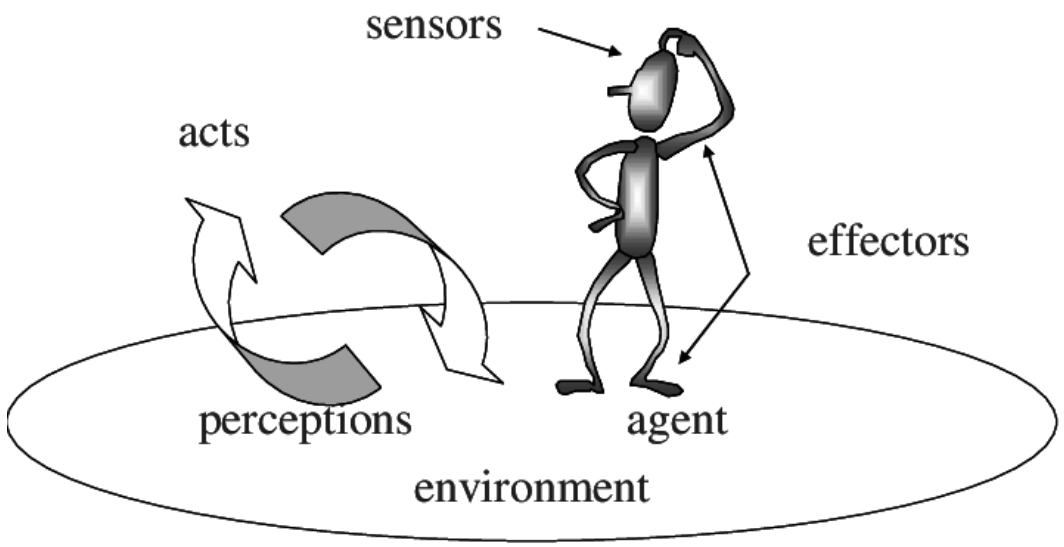
北京大学 PEKING UNIVERSITY

- 查找、导航、回忆、解决问题等等均与搜索相关

- 当一个机器知道如何规划自己接下来的决策/动作了，那这个机器可以叫有智能吗?

回忆：啥是智能体

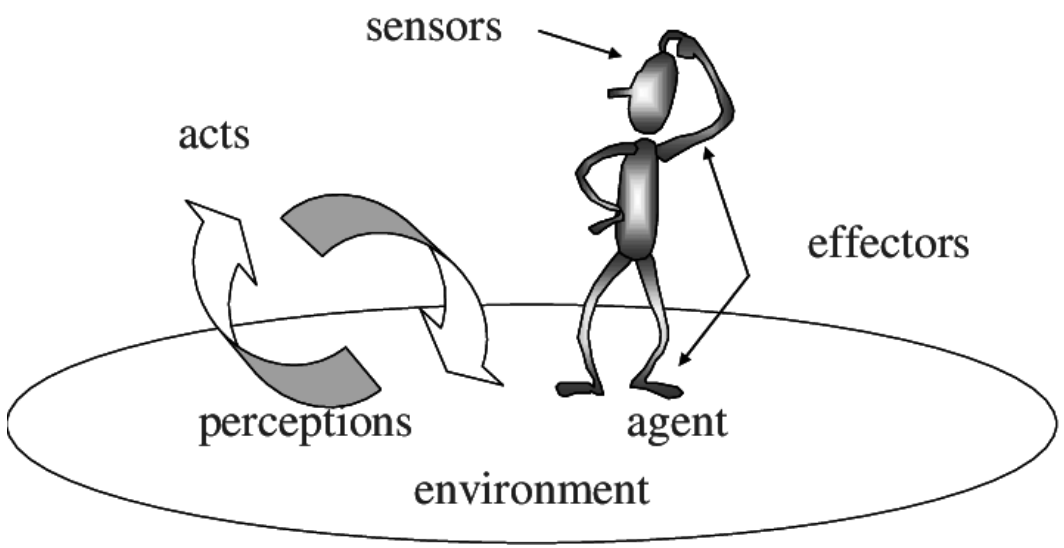
- 任何通过**传感器**(sensor)**感知**(percept)**环境**(environment)并通过**执行器**(actuator/effectors)作用于该环境的事物都可以被视为**智能体**



什么是搜索

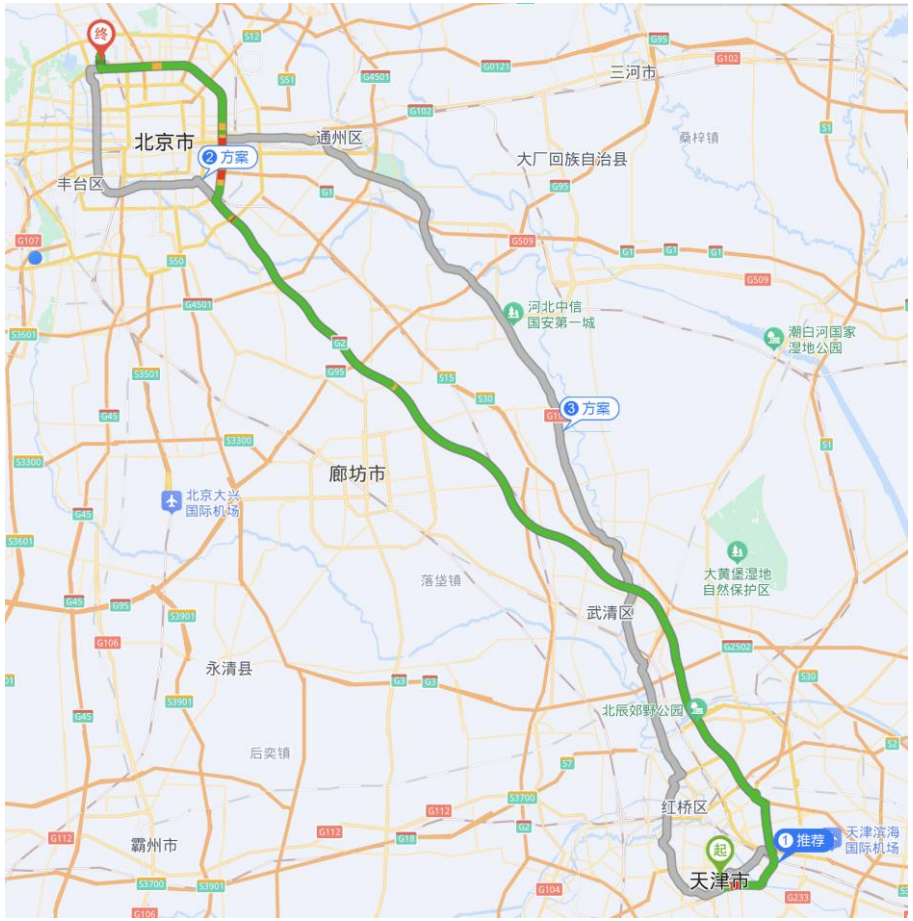
北京大学 PEKING UNIVERSITY

- 回忆一下我们今天是怎么来到这间教室，或者说节后怎么回到校园的?



什么是搜索

- 节后怎么回到校园的?



什么是搜索

北京大学 PEKING UNIVERSITY

- 如何让曹操离开重重包围?



什么是搜索

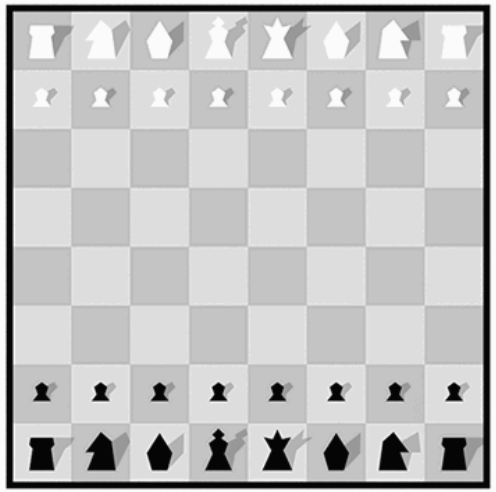
北京大学 PEKING UNIVERSITY

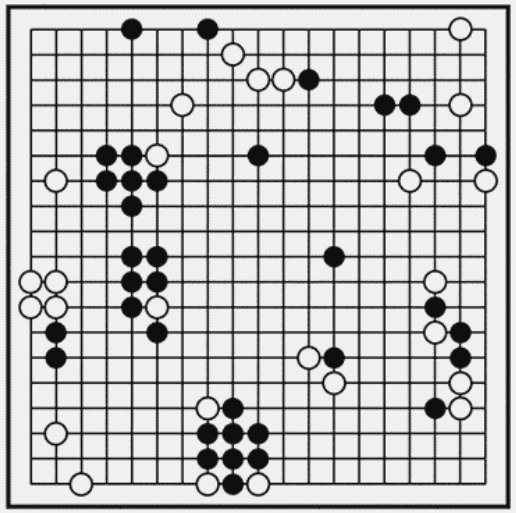
- 如何离开一个迷宫?



什么是搜索

- 各类棋类?





什么是搜索

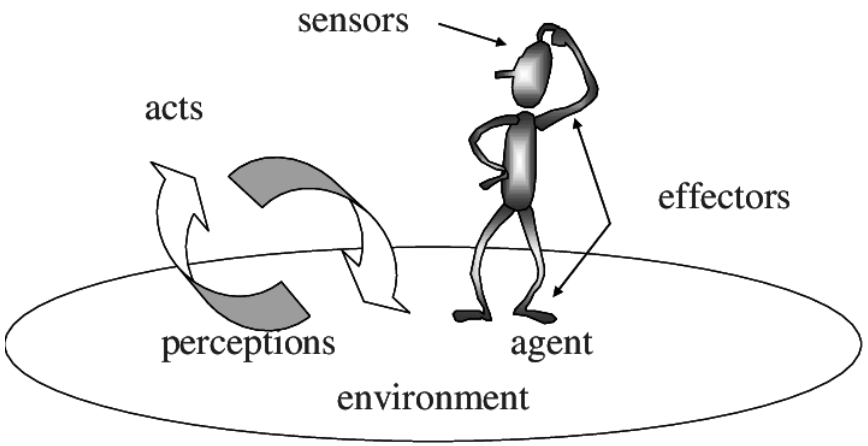
北京大学 PEKING UNIVERSITY

- 任何智能问题都可以转为搜索问题？比如：答案可以看作所有可能文本/图像空间的一个点
- 大模型的输出：可以模块化地变成搜索树

什么是搜索

- 根据Wikipedia：In computer science, a search algorithm is an algorithm designed to solve a search problem.

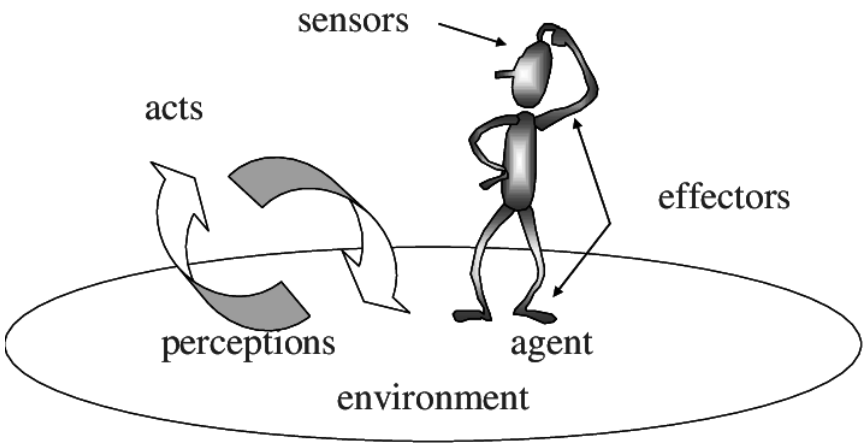
什么是搜索



定义部分可以参考书籍《Artificial Intelligence：A Modern Approach》第二部分problem-solving。

1. **目标（goal）：即我们要去“找”什么；什么时候可以结束搜索**

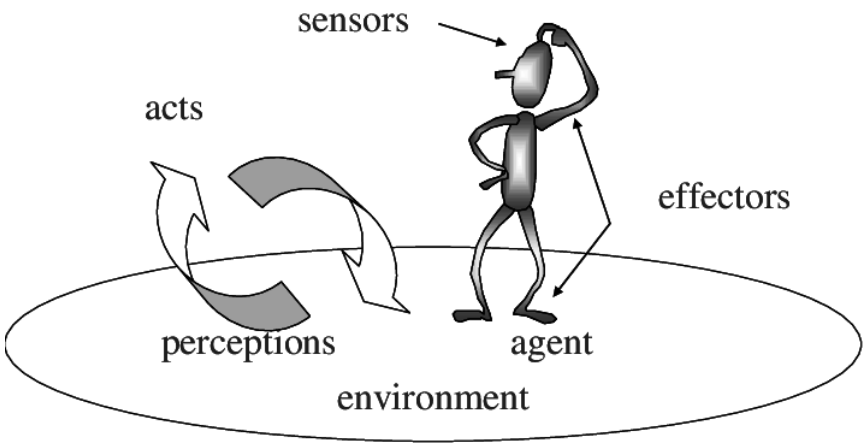
什么是搜索



定义部分可以参考我们的参考书籍《Artificial Intelligence：A Modern Approach》第二部分problem-solving。

1. **目标（goal）：即我们要去“找”什么；什么时候可以结束搜索**
2. **状态（state）：这里面其实包括三个主要的部分，开始状态（initial states），目标状态（goal states），当前状态（current state）**

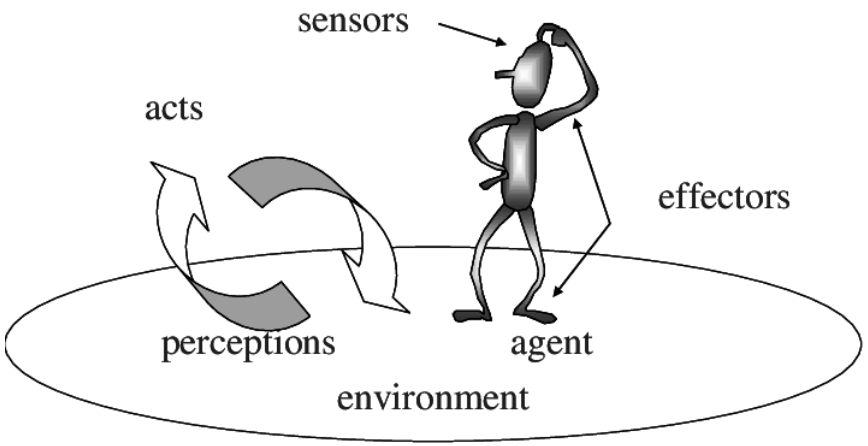
什么是搜索



定义部分可以参考我们的参考书籍《Artificial Intelligence：A Modern Approach》第二部分problem-solving。

1. **目标（goal）：即我们要去“找”什么；什么时候可以结束搜索**
2. **状态（state）：这里面其实包括三个主要的部分，开始状态（initial states），目标状态（goal states），当前状态（current state）**
3. **动作（actions）：智能体可以采取的行动/决策**

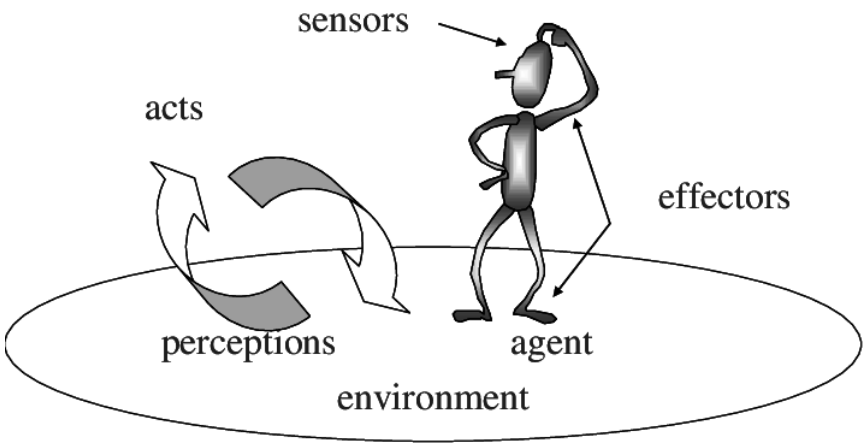
什么是搜索



定义部分可以参考我们的参考书籍《Artificial Intelligence：A Modern Approach》第二部分problem-solving。

1. **目标（goal）：即我们要去“找”什么；什么时候可以结束搜索**
2. **状态（state）：这里面其实包括三个主要的部分，开始状态（initial states），目标状态（goal states），当前状态（current state）**
3. **动作（actions）：智能体可以采取的行动/决策**
4. **（状态）转移方程（transition function）：当前状态随着动作会怎么变化。**

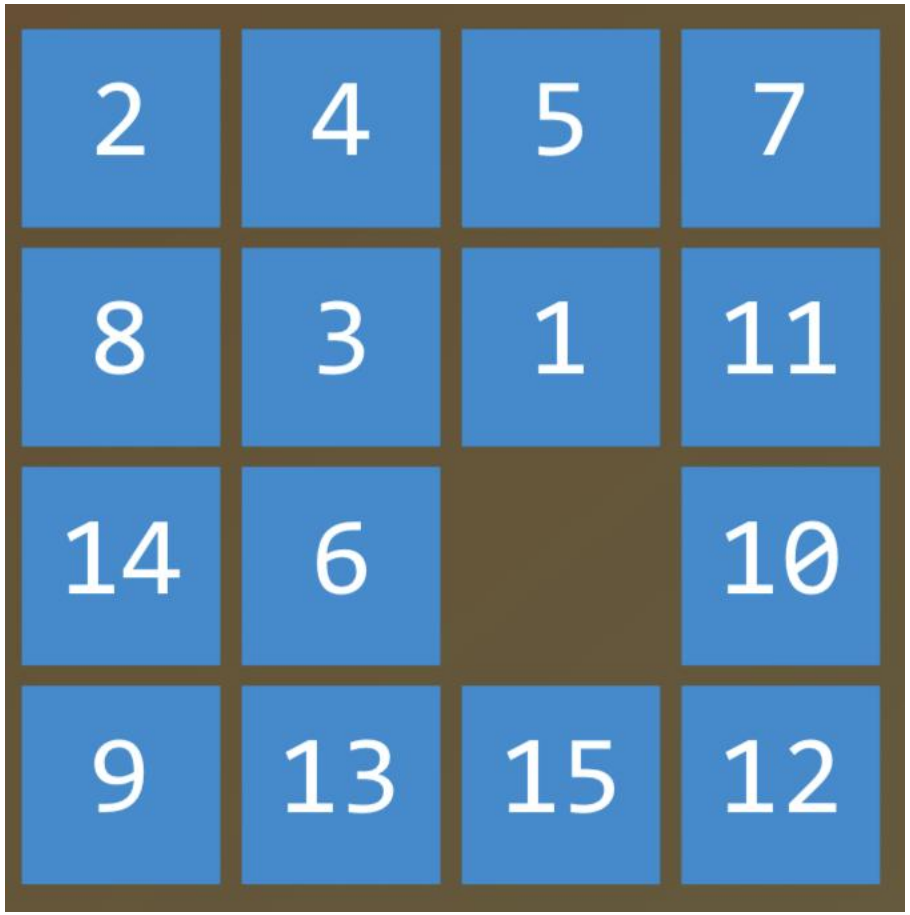
什么是搜索



定义部分可以参考我们的参考书籍《Artificial Intelligence：A Modern Approach》第二部分problem-solving。

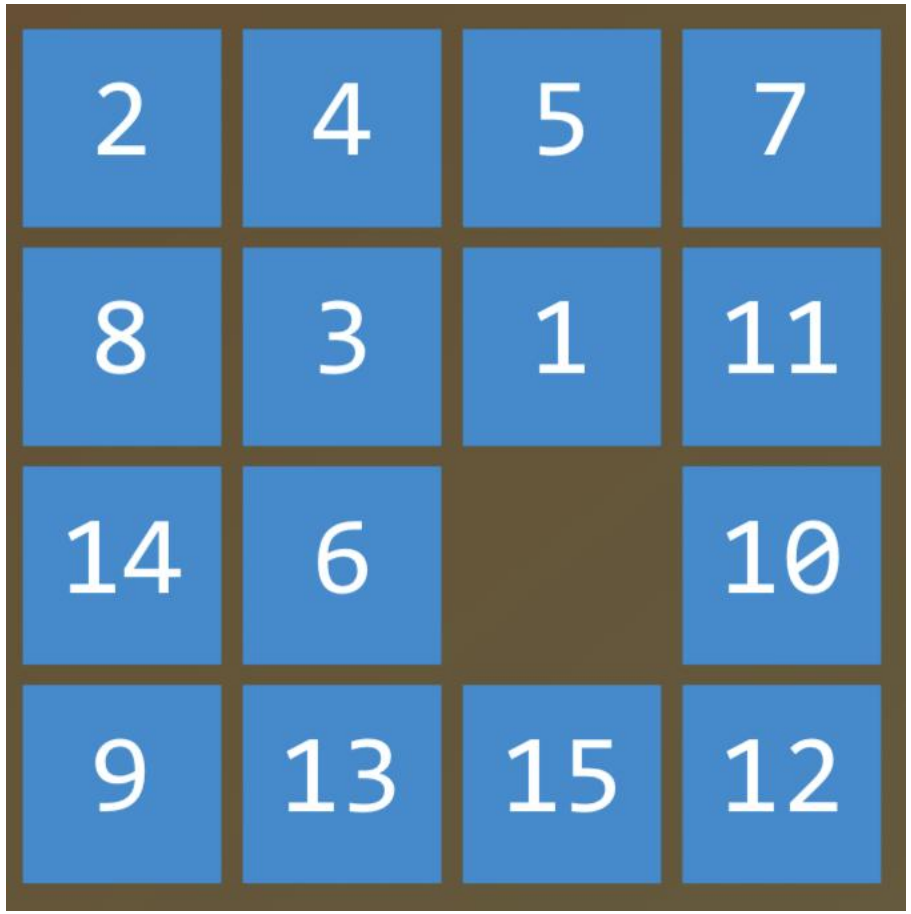
1. **目标（goal）：即我们要去“找”什么；什么时候可以结束搜索**
2. **状态（state）：这里面其实包括三个主要的部分，开始状态（initial states），目标状态（goal states），当前状态（current state）**
3. **动作（actions）：智能体可以采取的行动/决策**
4. **（状态）转移方程（transition function）：当前状态随着动作会怎么变化**
5. **成本/代价方程（cost function）：每个动作要消耗多大的成本/代价**

举个例子

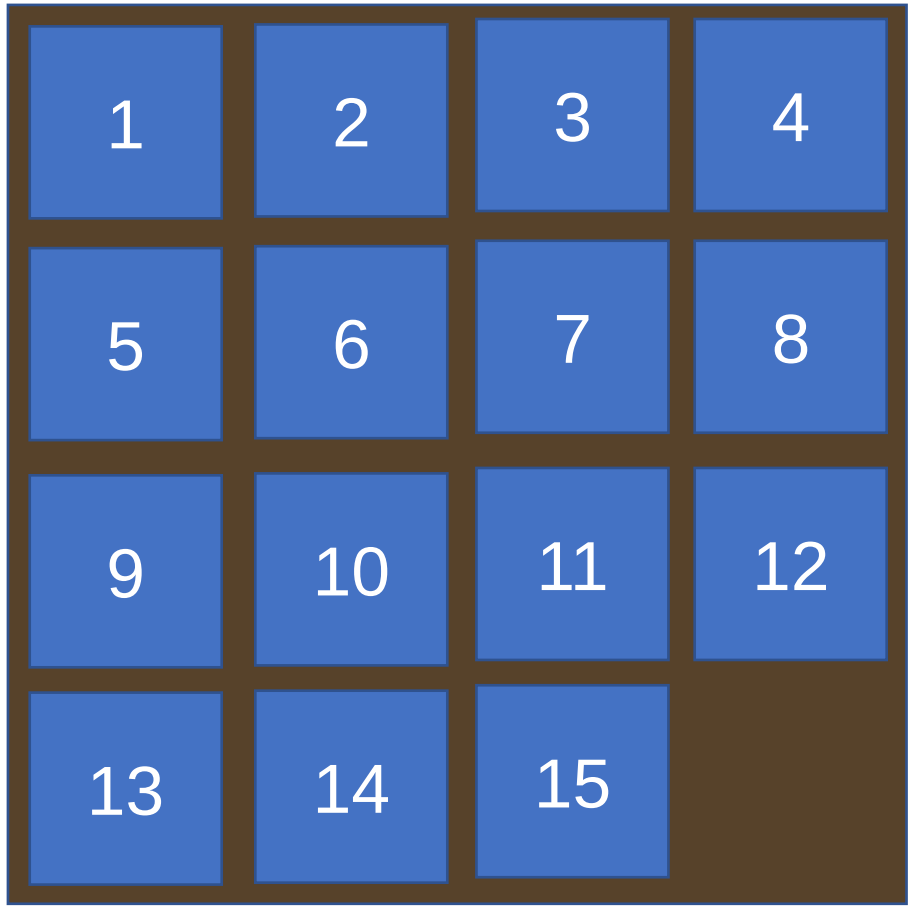


如何将方块从小到大排列?

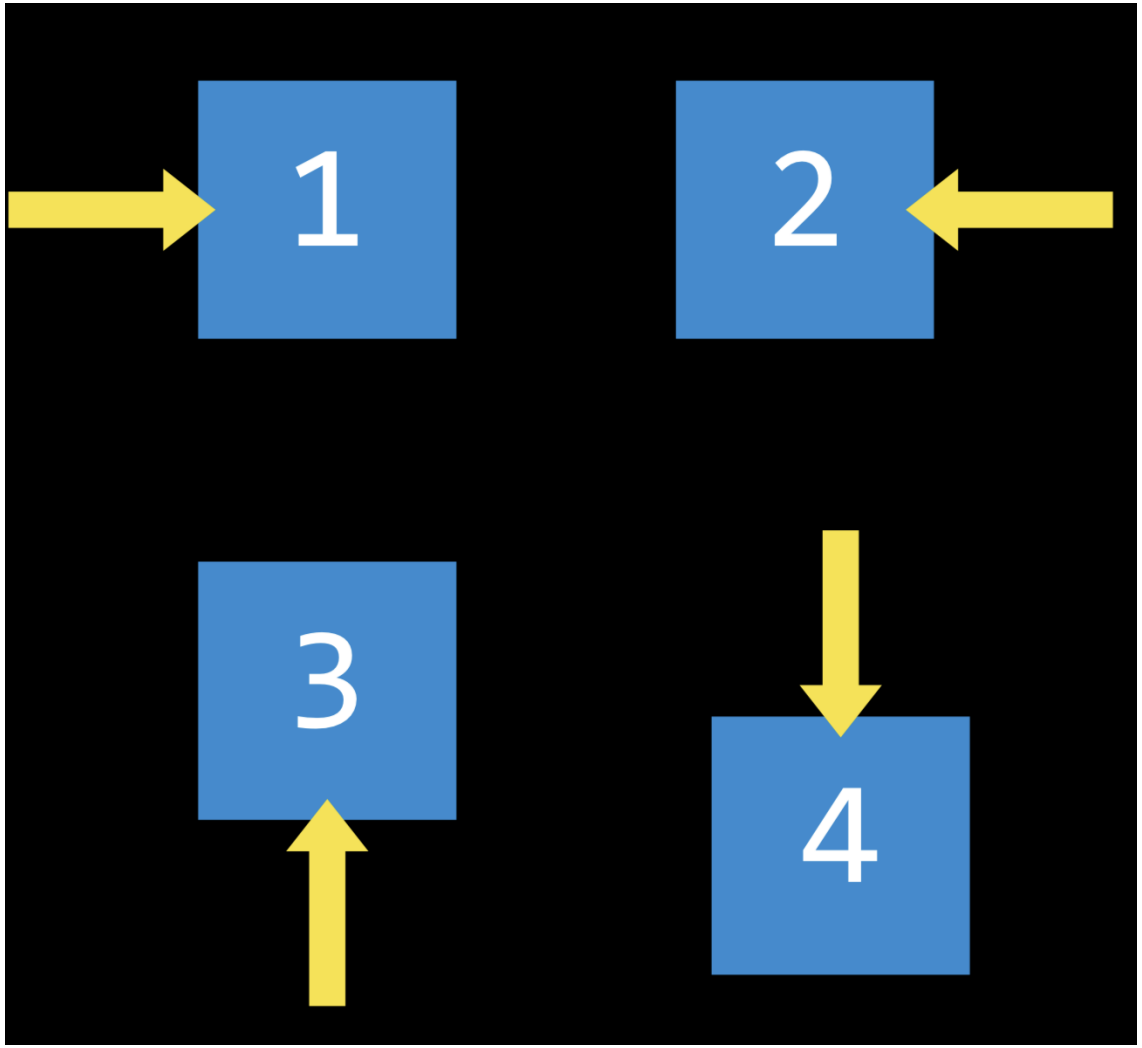
举个例子：开始状态



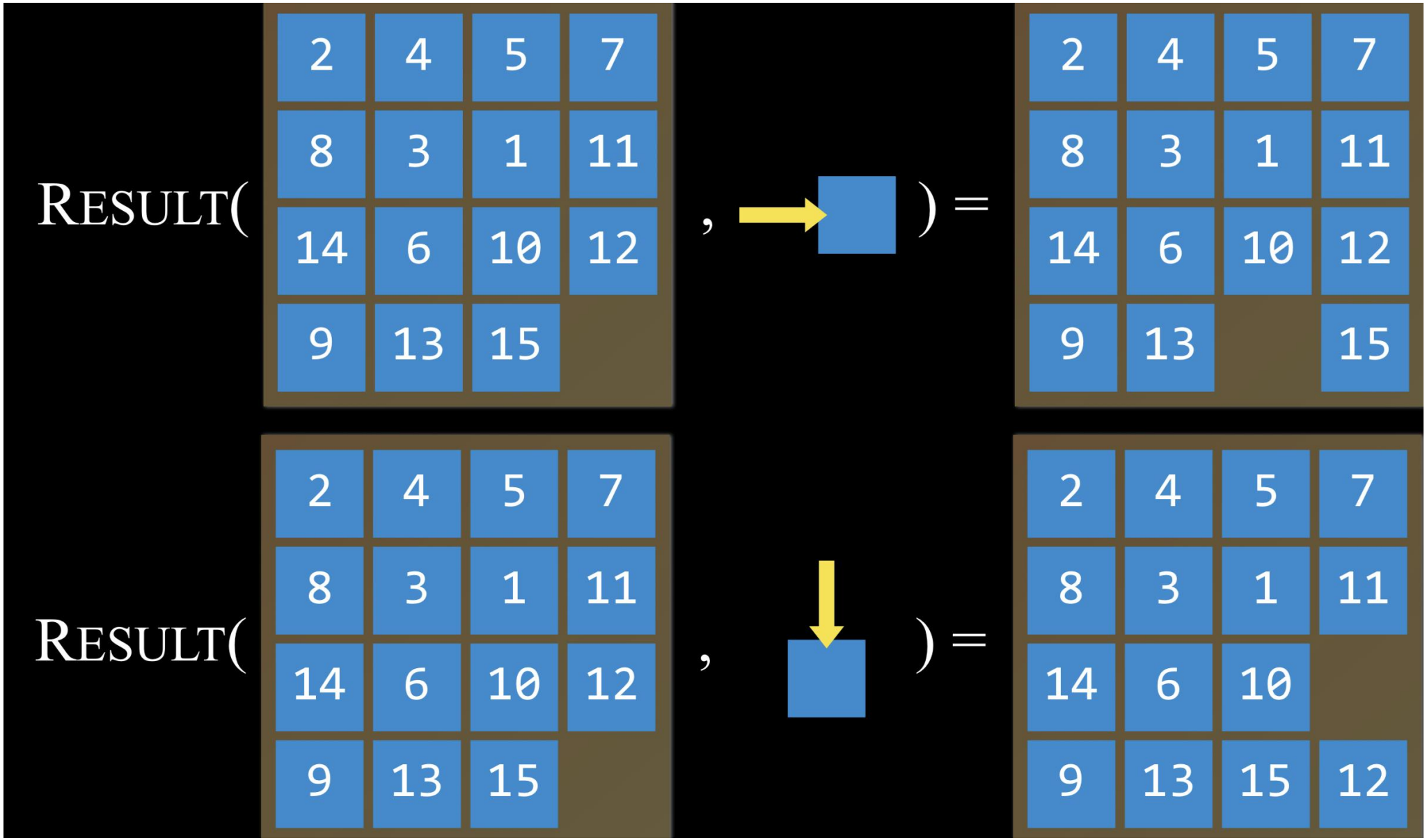
举个例子：目标状态



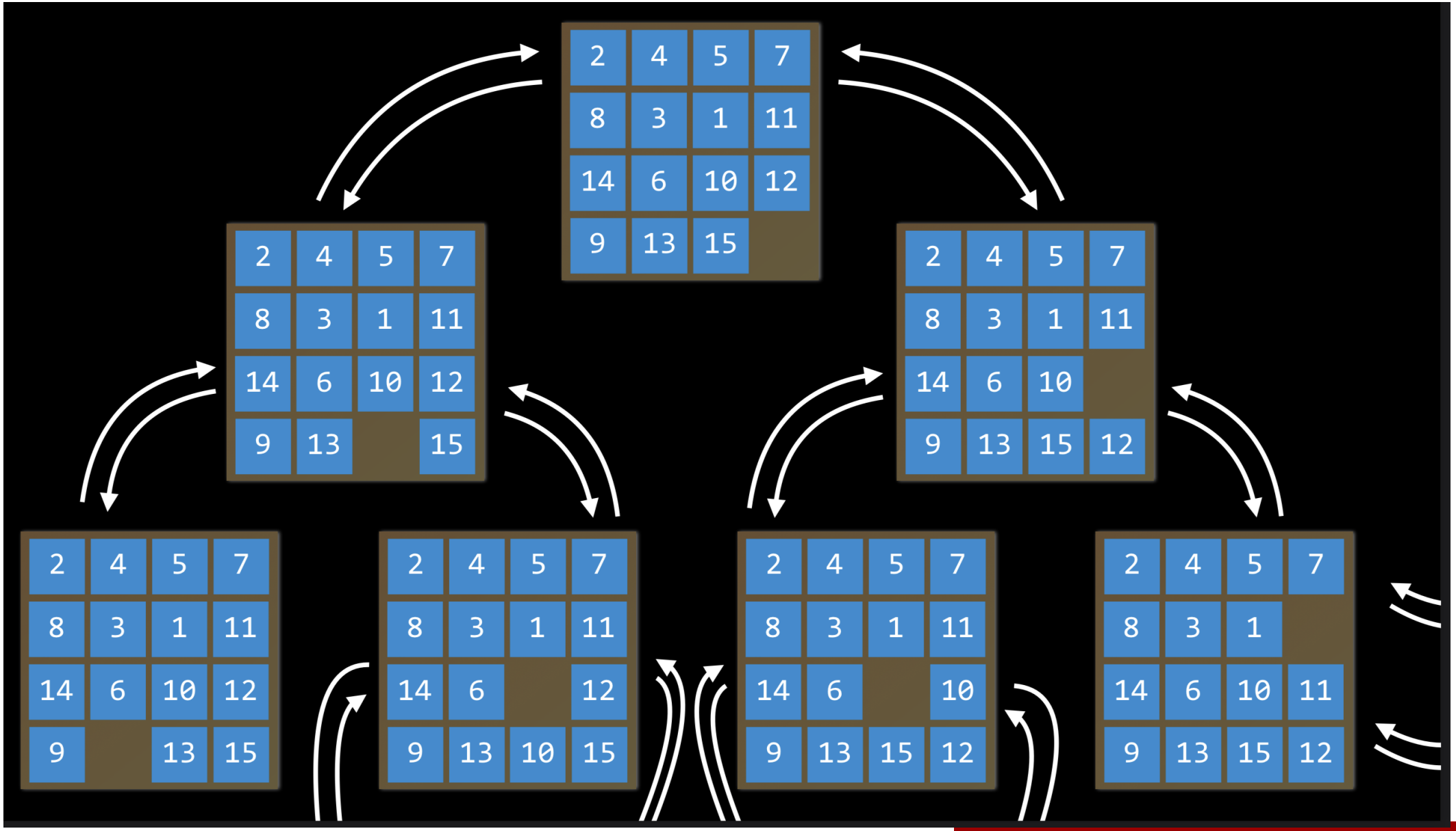
举个例子：动作



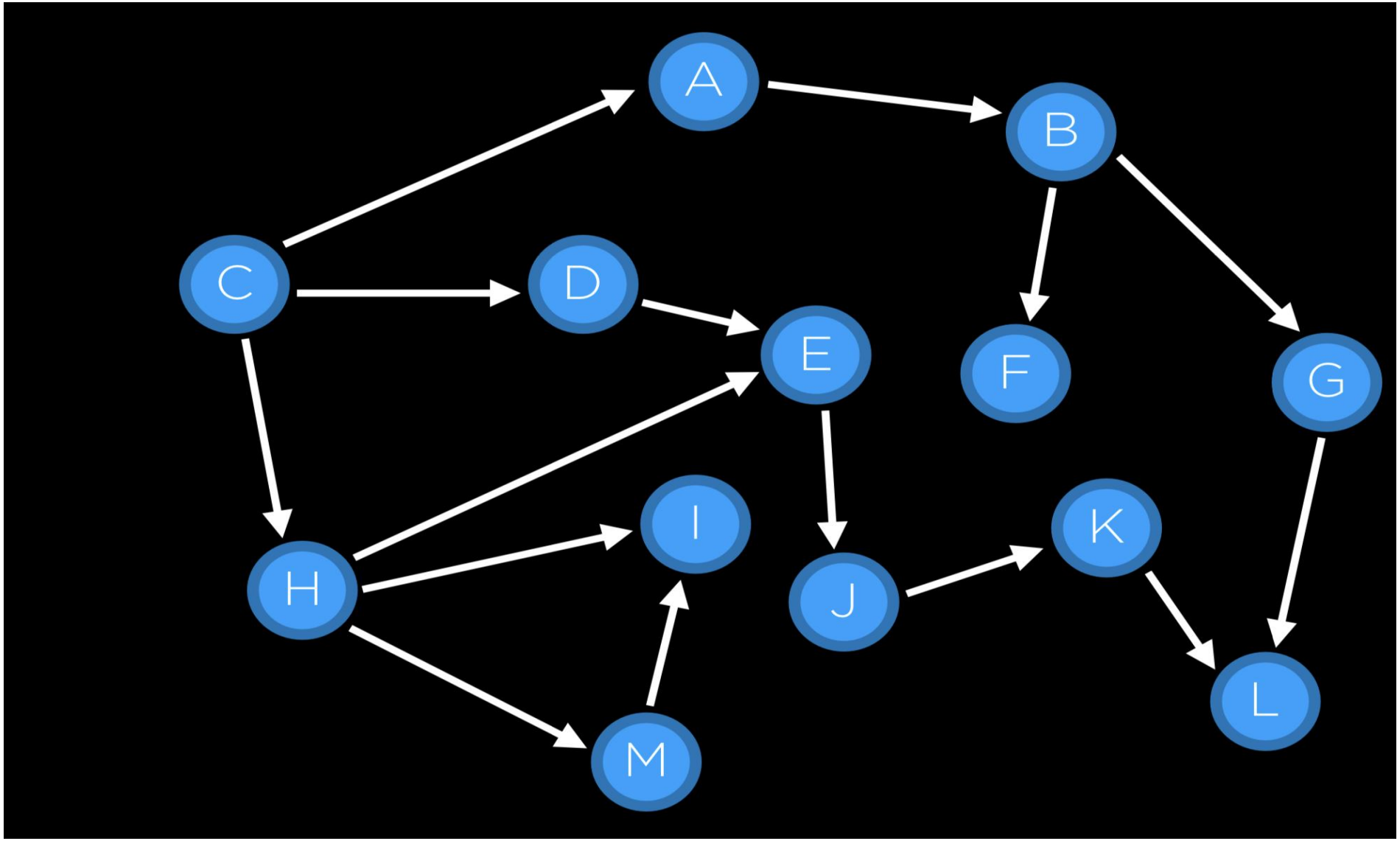
举个例子：转移方程


RESULT(
2 4 5 7
8 3 1 11
14 6 10 12
9 13 15
,
)=
2 4 5 7
8 3 1 11
14 6 10 12
9 13 15
RESULT(
2 4 5 7
8 3 1 11
14 6 10 12
9 13 15
,
)=
2 4 5 7
8 3 1 11
14 6 10
9 13 15 12

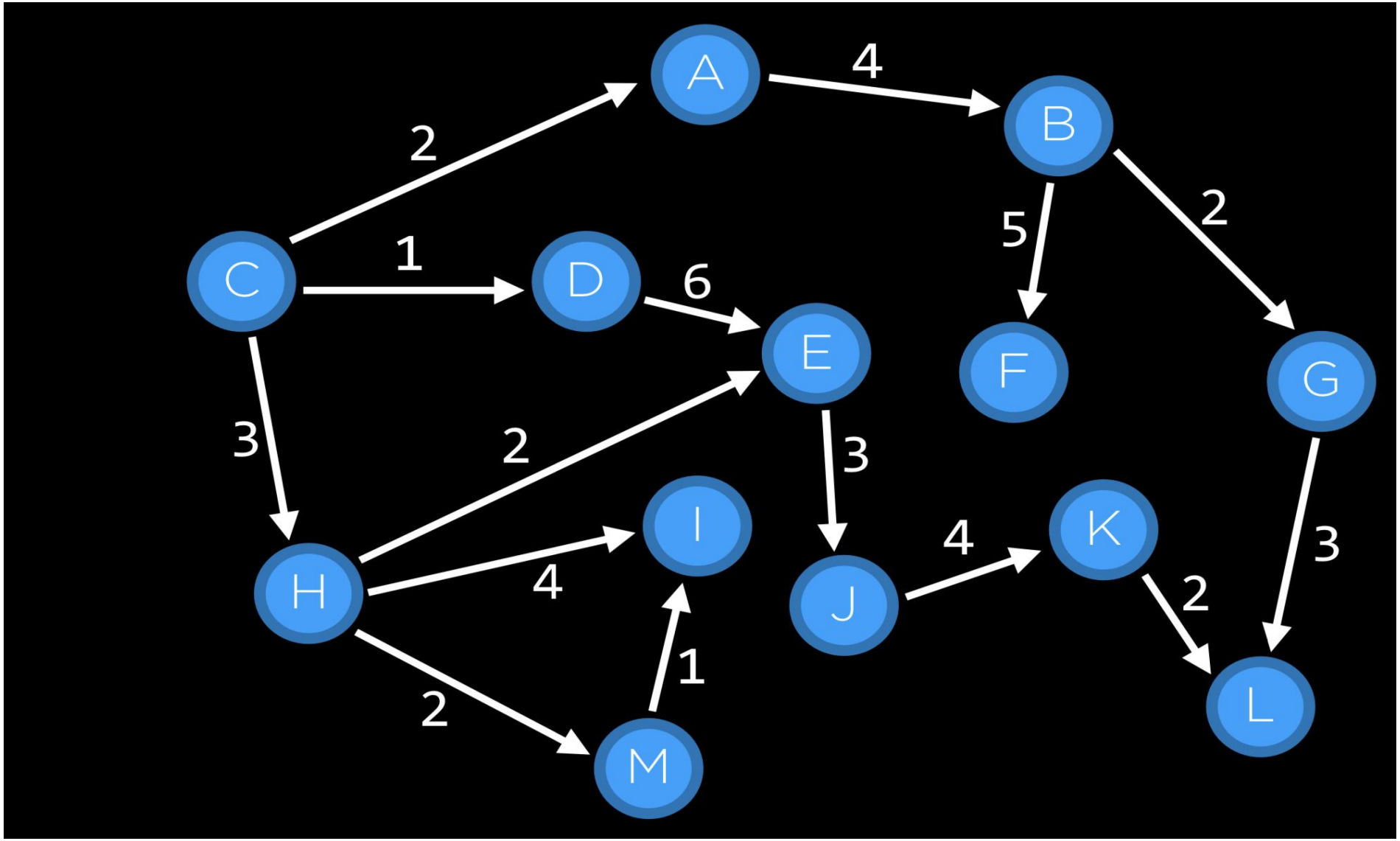
举个例子：状态，动作，转移


2 4 5 7
8 3 1 11
14 6 10 12
9 13 15
2 4 5 7
8 3 1 11
14 6 10 12
9 13 15
2 4 5 7
8 3 1 11
14 6 10
9 13 15 12
2 4 5 7
8 3 1 11
14 6 10 12
9 13 15
2 4 5 7
8 3 1 11
14 6 12
9 13 10 15
2 4 5 7
8 3 1 11
14 6 10
9 13 15 12
2 4 5 7
8 3 1
14 6 10 11
9 13 15 12

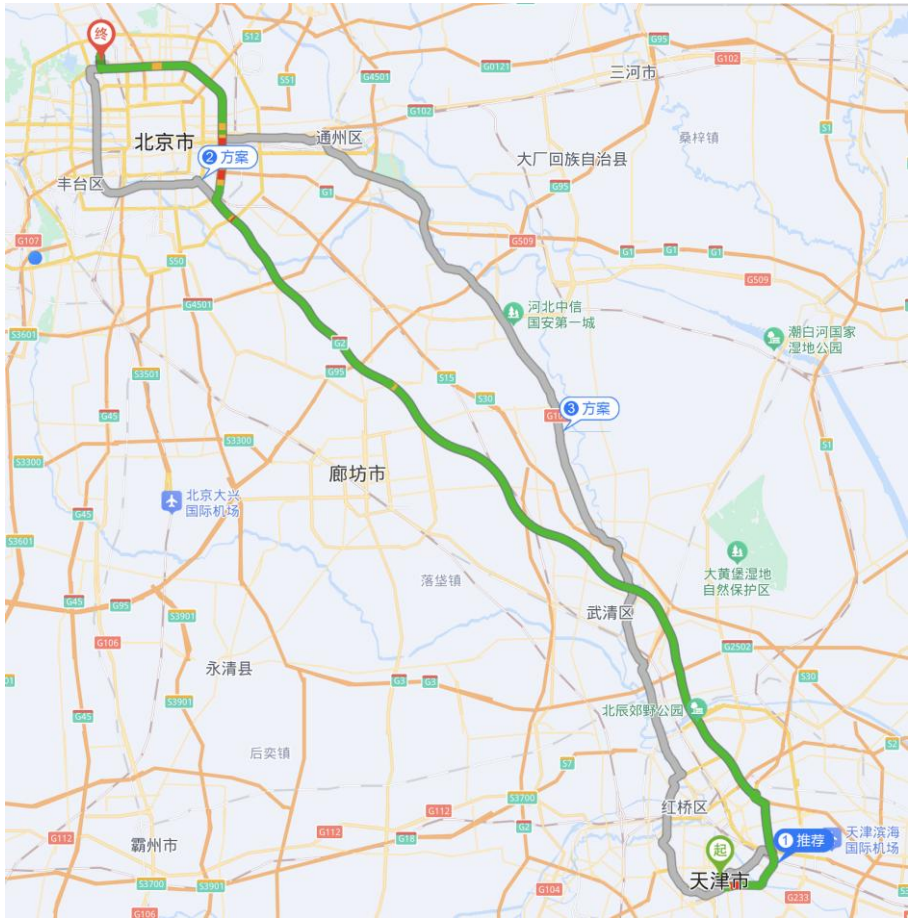
举个例子：状态，动作，转移



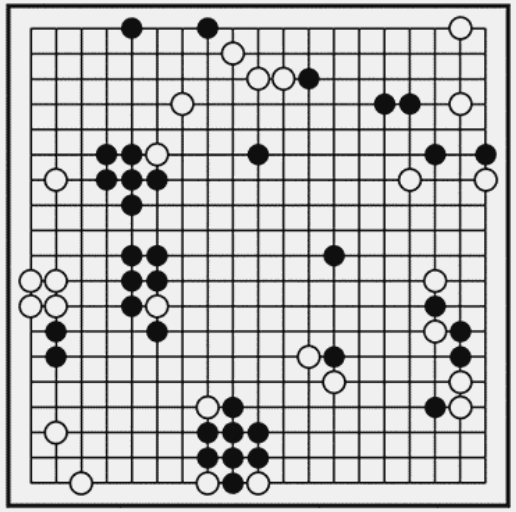
举个例子：成本/代价方程



练习


终
北京市
丰台区
通州区
三河市
大厂回族自治县
桑梓镇
② 方案
③ 方案
河北中信
国安第一城
潮白河国家
湿地公园
廊坊市
北京大兴
国际机场
落垡镇
大黄堡湿地
自然保护区
武清区
永清县
北辰郊野公园
后奕镇
红桥区
霸州市
① 推荐
起
天津滨海
国际机场
天津市

练习

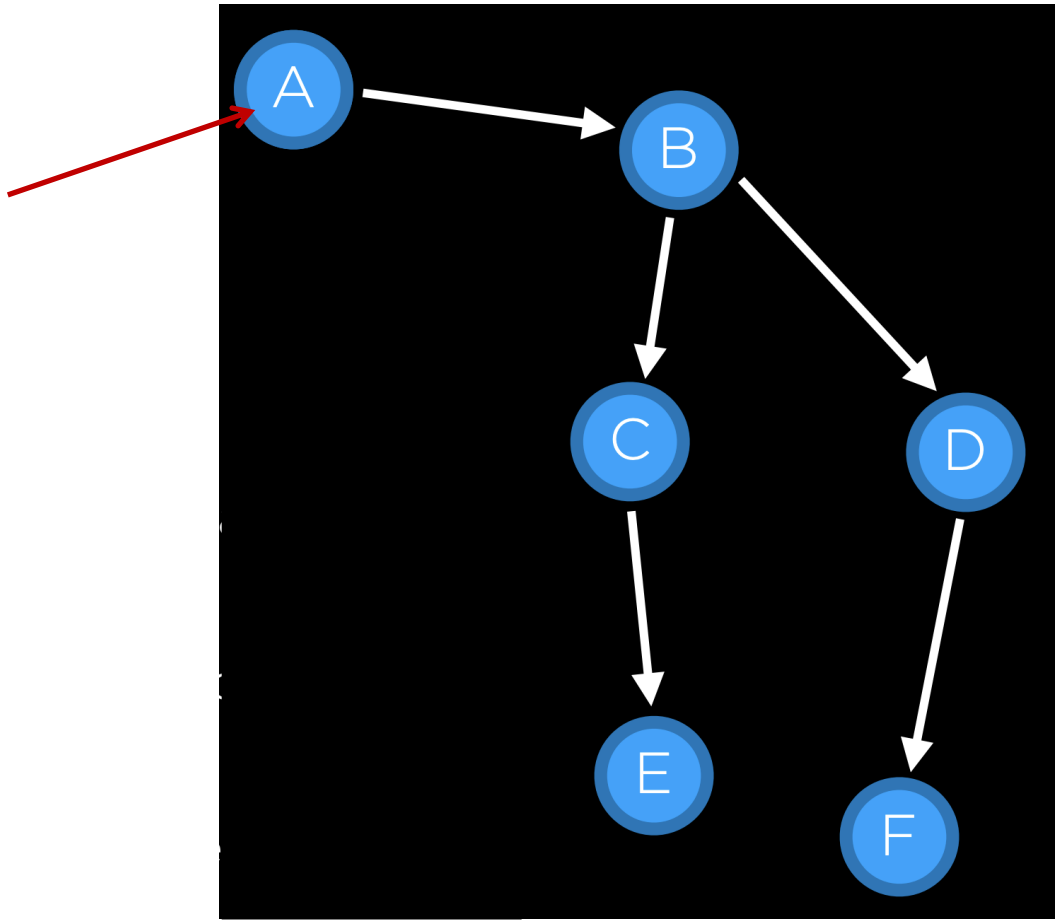


怎么解决一个搜索问题

北京大学 PEKING UNIVERSITY

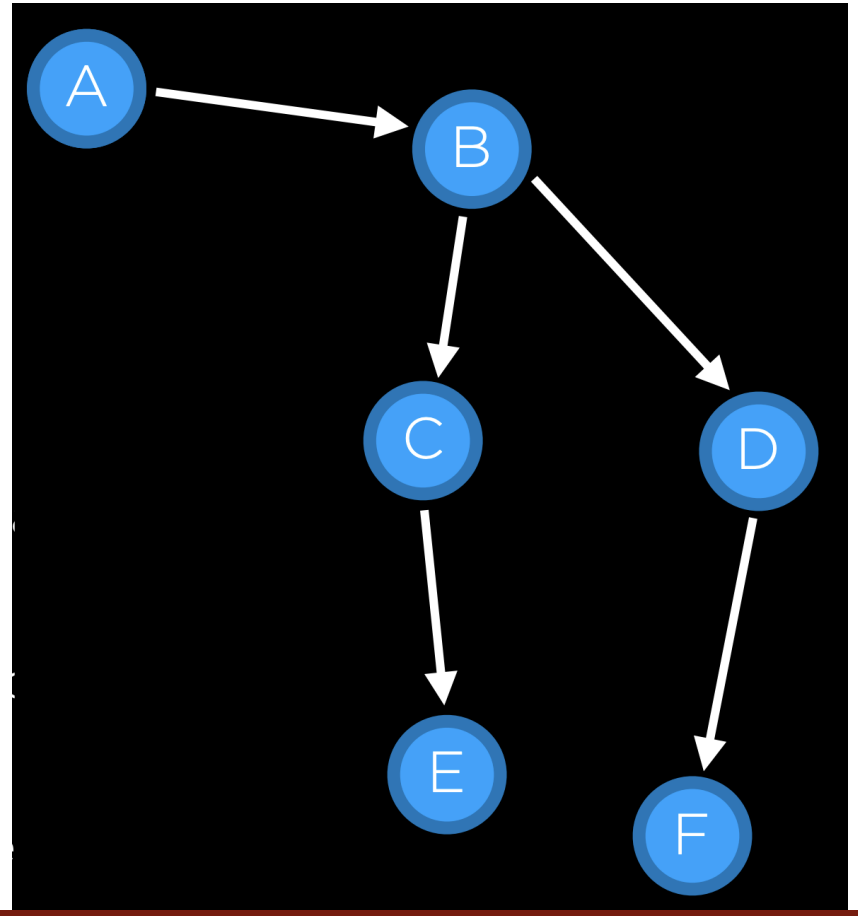
- 一个搜索问题的答案长什么样?

怎么解决一个搜索问题



1. 构建一个列表，表示我们现在可以考虑的状态。
2. 如果列表为空，表示我们失败了。
3. 如果列表里包含某个目标状态，表示我们成功了。
4. 从列表移除一个节点。这就是我们的当前节点了。（注意移头还是移尾？）
5. 把某些和当前节点相关的节点，加入列表。（加某一个还是都加入？）

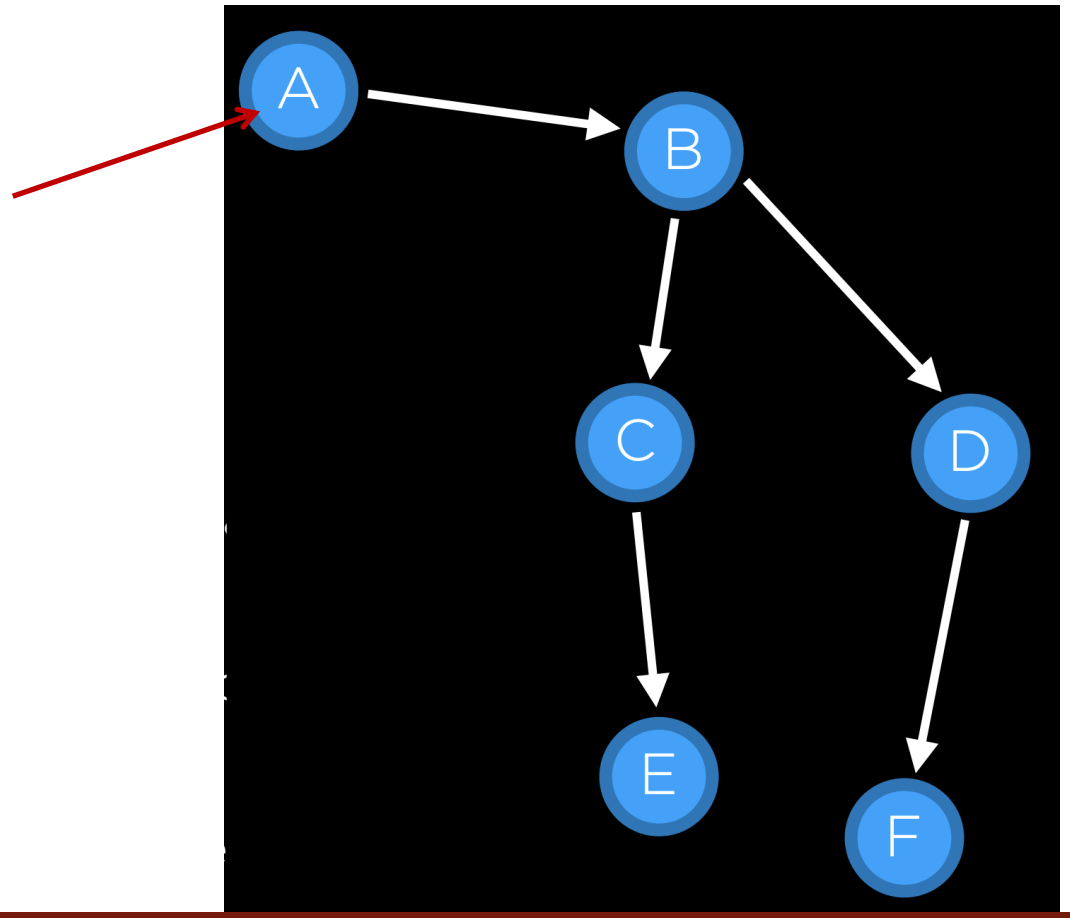
怎么解决一个搜索问题





1. 构建一个列表，表示我们现在可以考虑的状态。：我们先把开始状态放进来

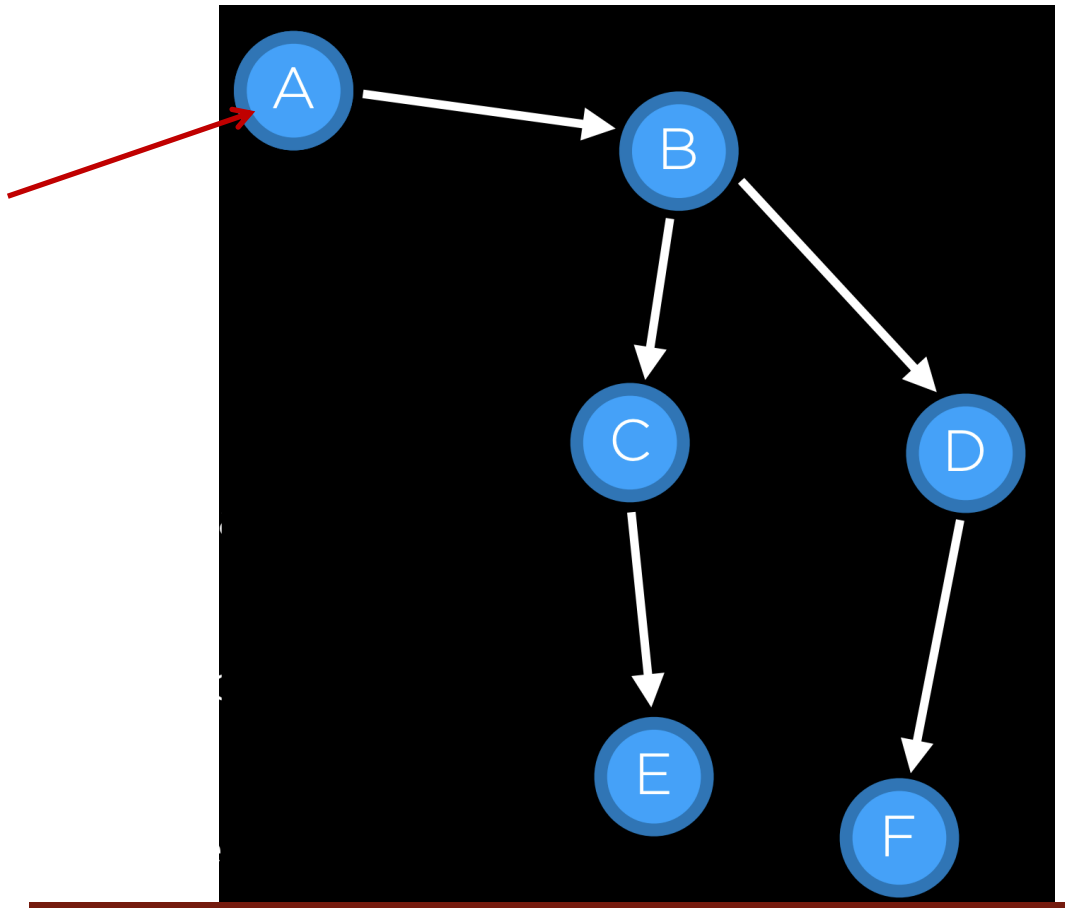
怎么解决一个搜索问题



4. 从列表移除一个节点。这就是我们的当前节点了。（注意移头还是移尾？）：

我们从表头移出一个节点，代表着我们当前状态来到了A

怎么解决一个搜索问题

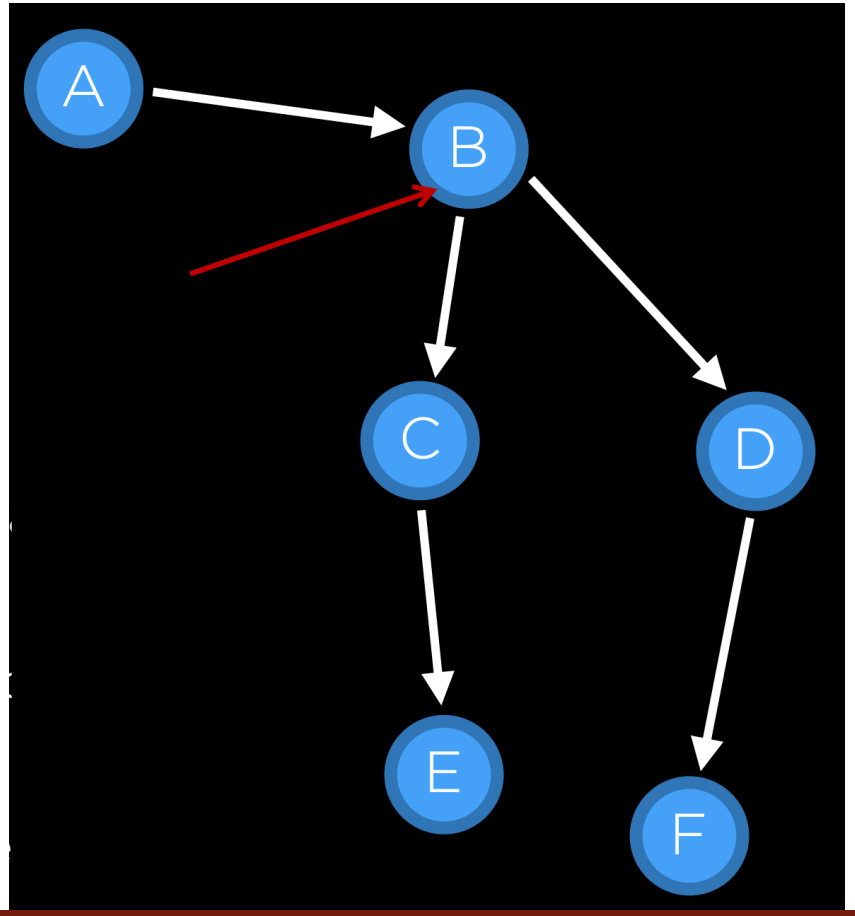


5. 把某些和当前节点相关的节点，加入列表。（加某一个还是都加入？）：

我们把和A一步就能到达的节点（B）放进来



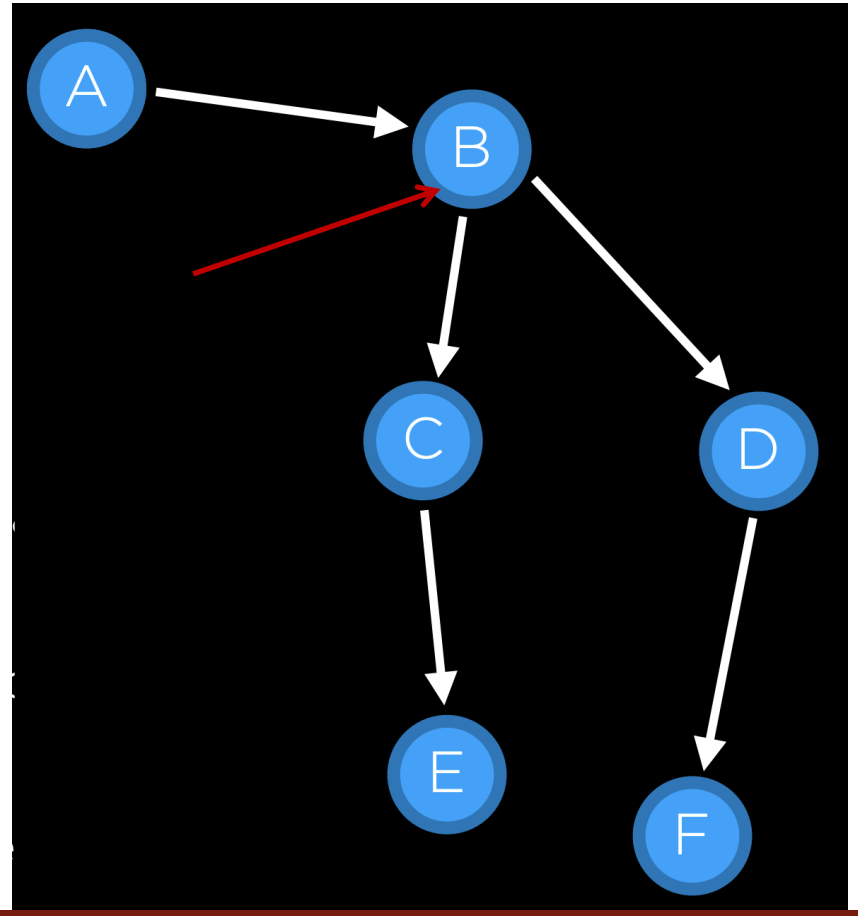
怎么解决一个搜索问题



4. 从列表移除一个节点。这就是我们的当前节点了。（注意移头还是移尾？）：

   我们从表头移出B，当前状态来到了B

怎么解决一个搜索问题

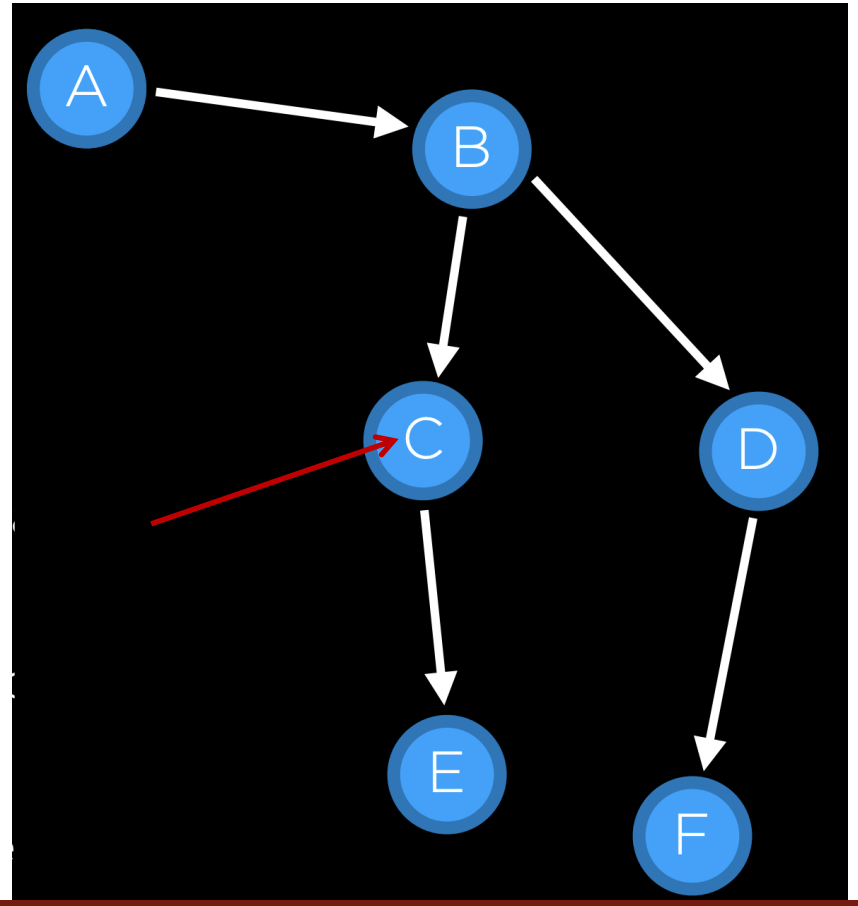


5. 把某些和当前节点相关的节点，加入列表。
（加某一个还是都加入？）：

把B一步能到达的节点（C、D）放进来



怎么解决一个搜索问题

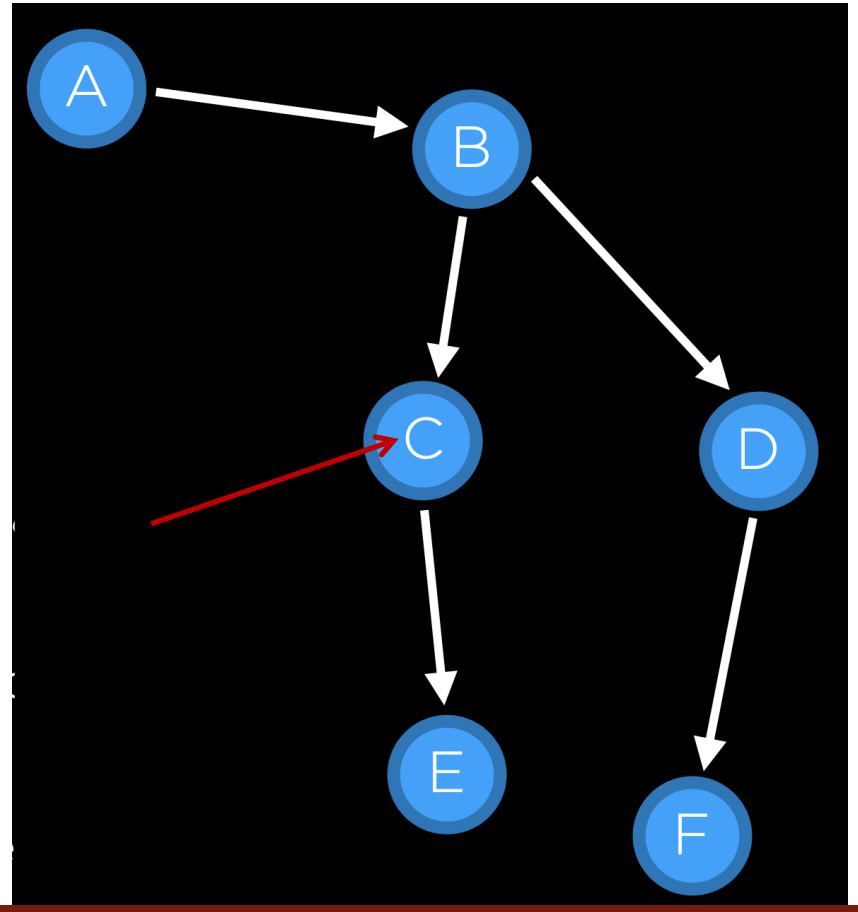


4. 从列表移除一个节点。这就是我们的当前节点了。（注意移头还是移尾？）：

从表头移出C，当前状态来到C



怎么解决一个搜索问题

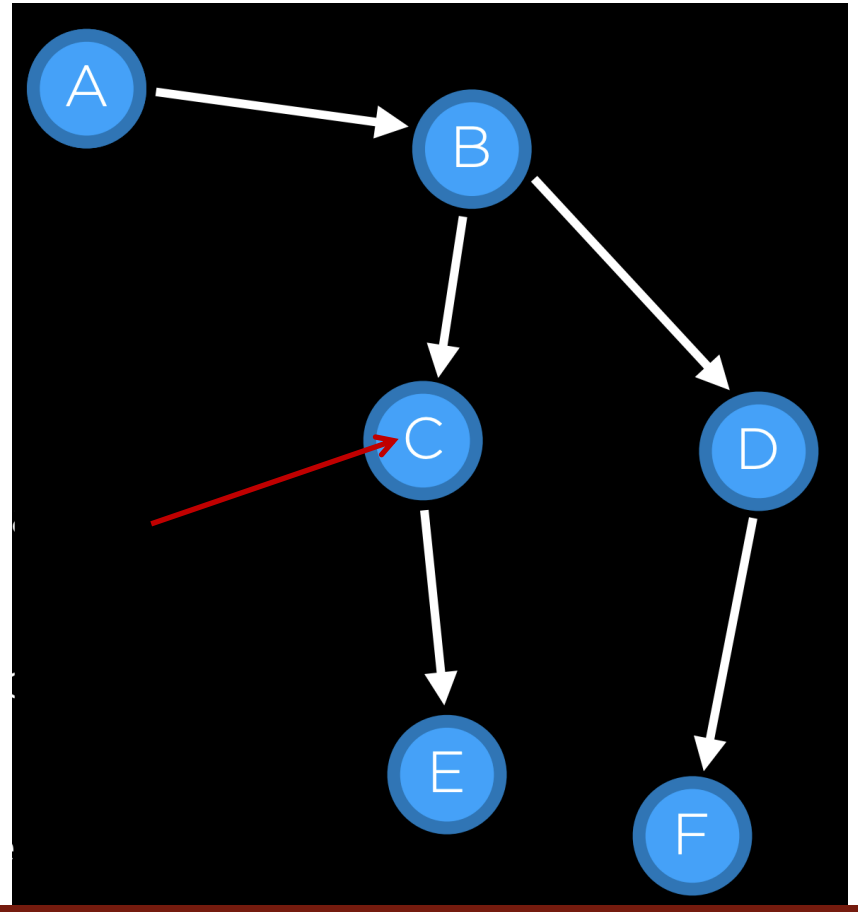




5. 把某些和当前节点相关的节点，加入列表。（加某一个还是都加入？）：

把C能一步到达的节点（E）加入

怎么解决一个搜索问题

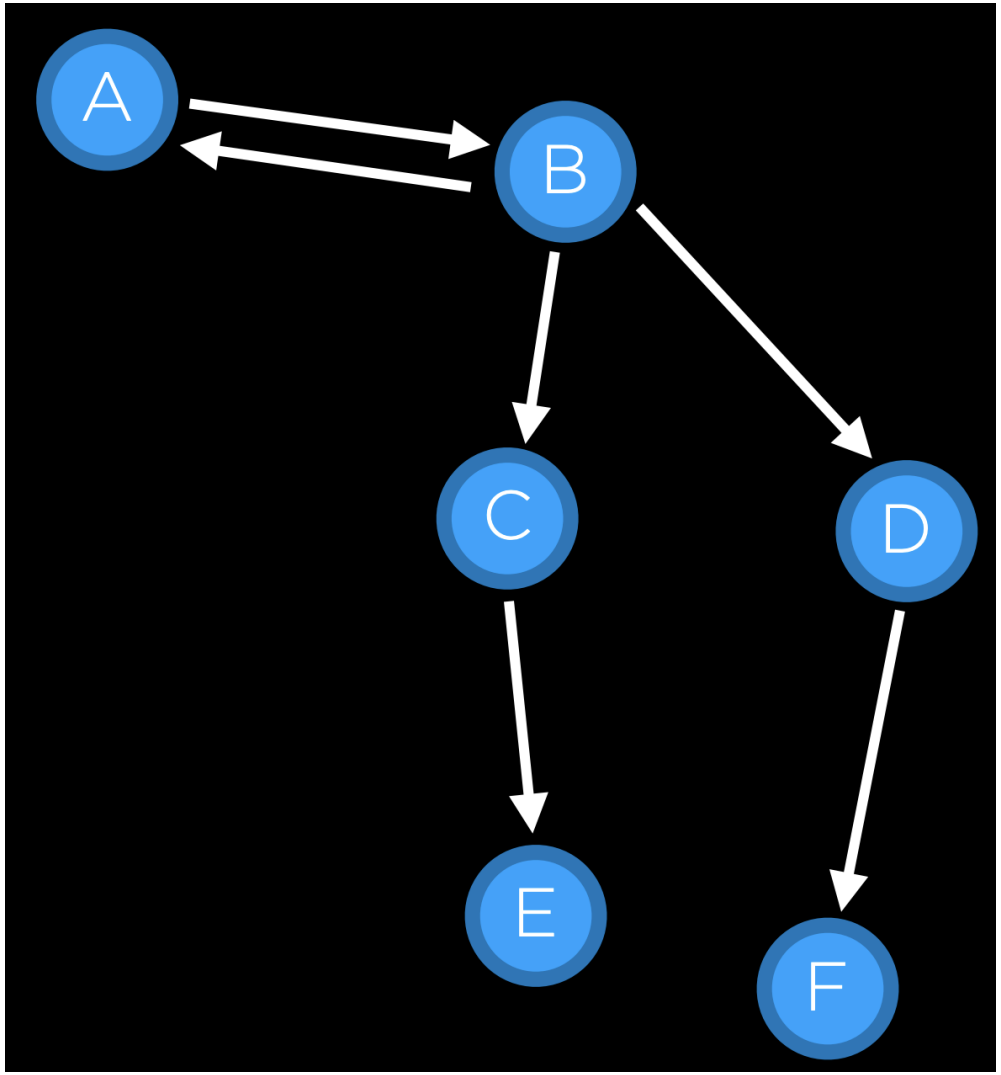




3. 如果列表里包含某个目标状态，表示我们成功了:

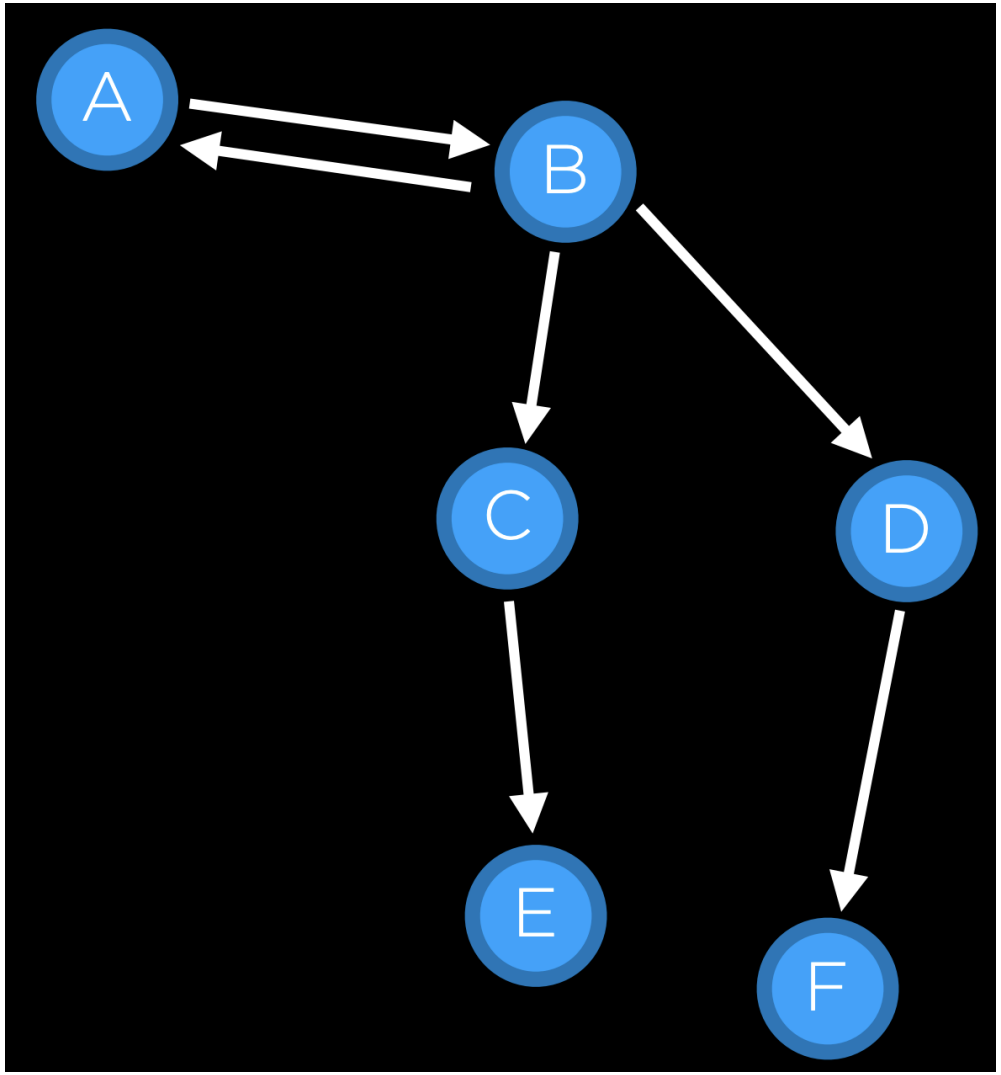
E是目标状态，并且在我们的表里面。成功。

最容易出错的地方



重复回到同一个节点

最容易出错的地方



我们重复一下刚才的流程

表只有A

从表头移出A（我们来到A）

把A一步能到达的（B）加入

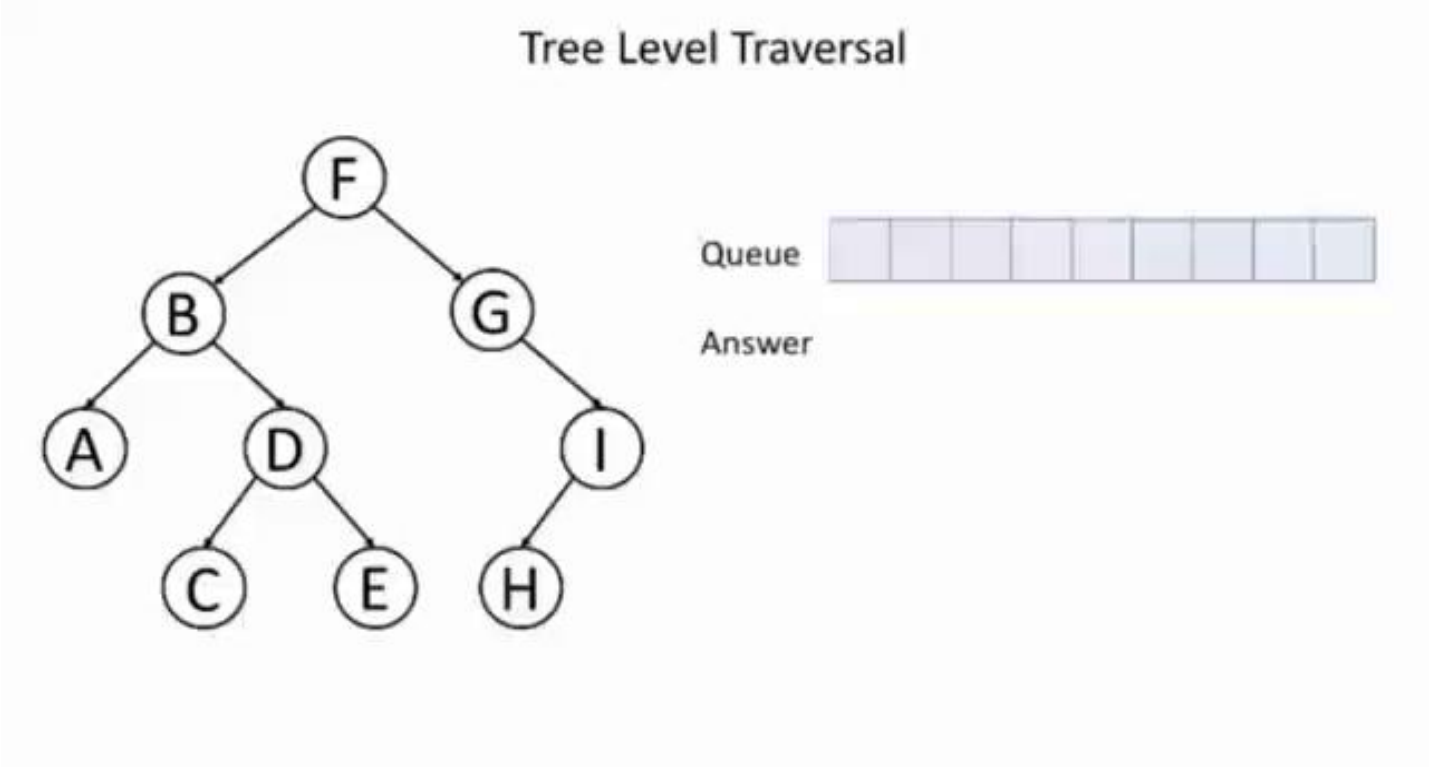
从表头移出B（我们来到B）

把B一步能到达的（A）加入

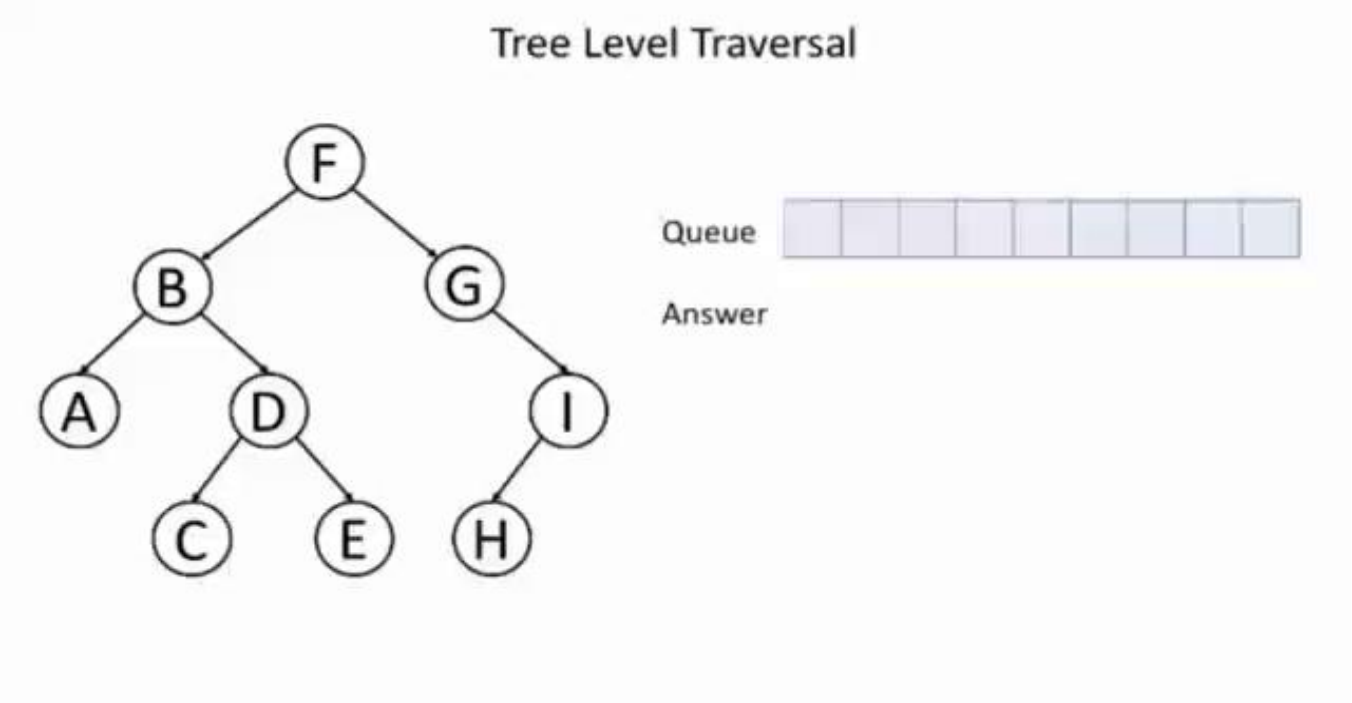
从表头移出A（我们来到A）

……

广度优先搜索



广度优先搜索



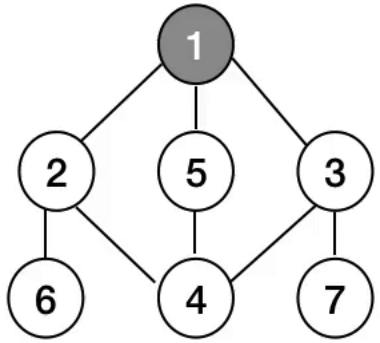
1. 首先将根节点放入*列表*中。

2. 从*列表*中取出第一个节点，并检验它是否为目标。
   如果找到目标，则结束搜寻并回传结果。
   否则将它所有尚未检验过的直接子节点加入队列中。

3. 若*列表*为空，表示整张图都检查过了——亦即图中没有欲搜寻的目标。结束搜寻并回传“找不到目标”。

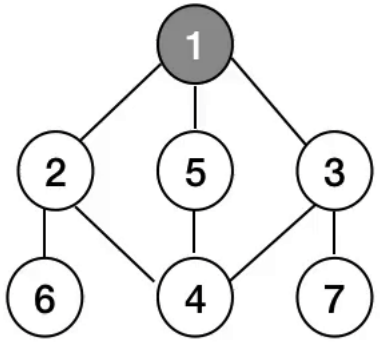
4. 重复步骤2。

深度优先搜索





深度优先搜索





1. 首先将根节点放入*列表*中。

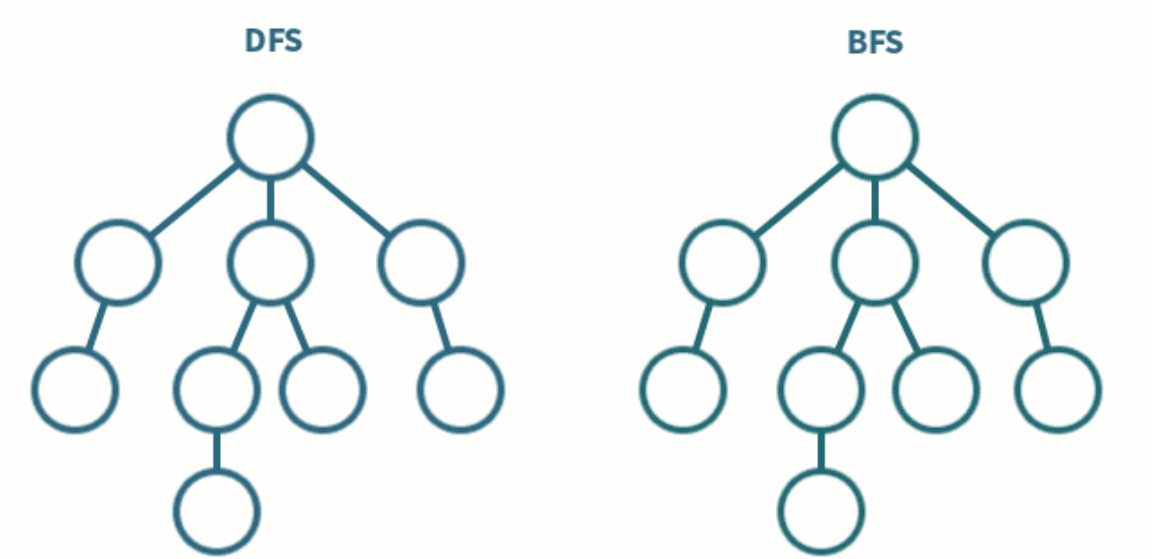
2. 关注*列表*中最后一个节点，并检验它是否为目标。
   如果找到目标，则结束搜寻并回传结果。
   否则将它某一个尚未检验过的直接子节点加入*列表*末尾。

3. 重复步骤2。

4. 如果不存在未检测过的直接子节点。
   删除当前节点。
   重复步骤2。

5. 若*列表*为空，表示整张图都检查过了——亦即图中没有欲搜寻的目标。结束搜寻并回传“找不到目标”

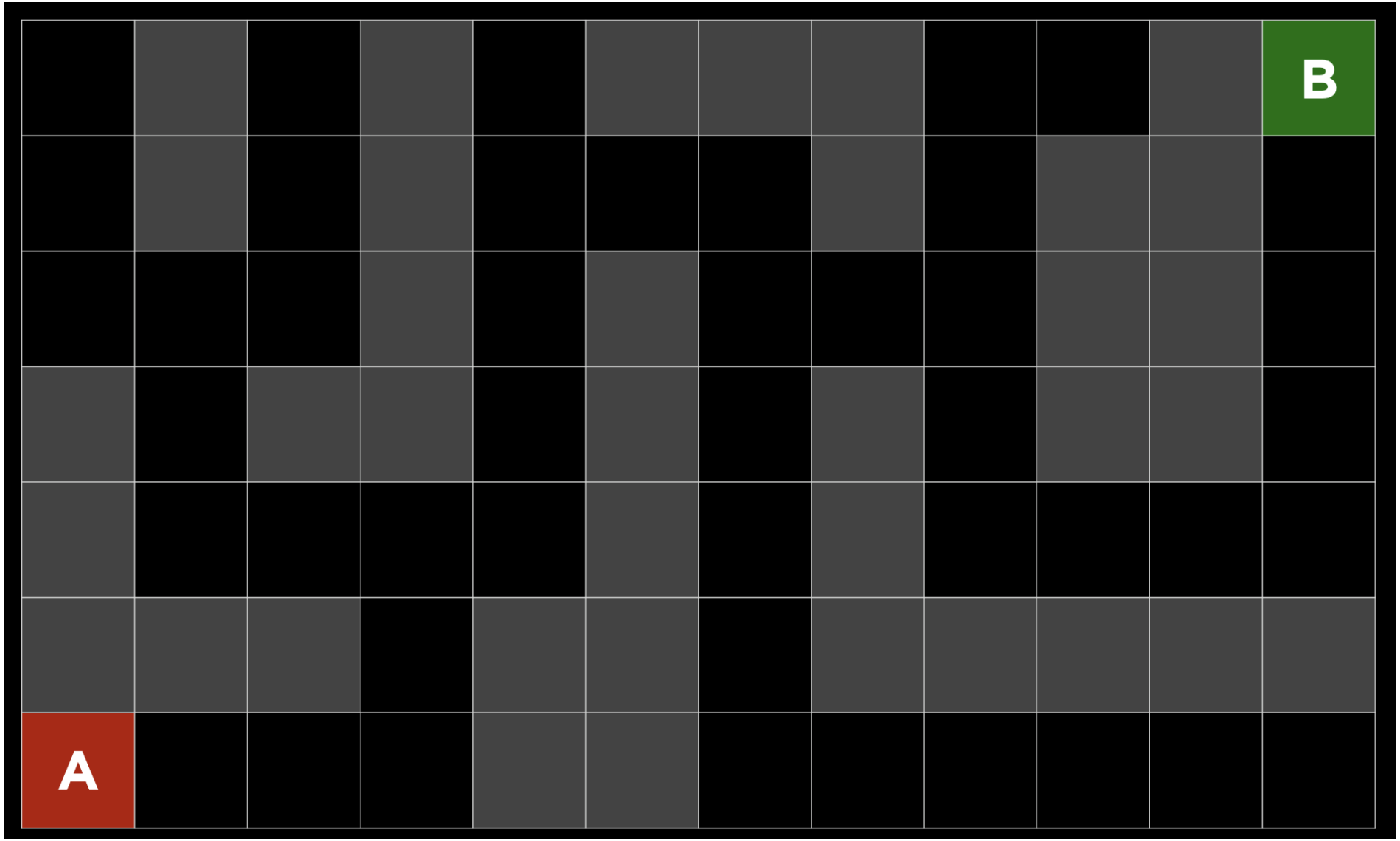
深搜 vs. 广搜



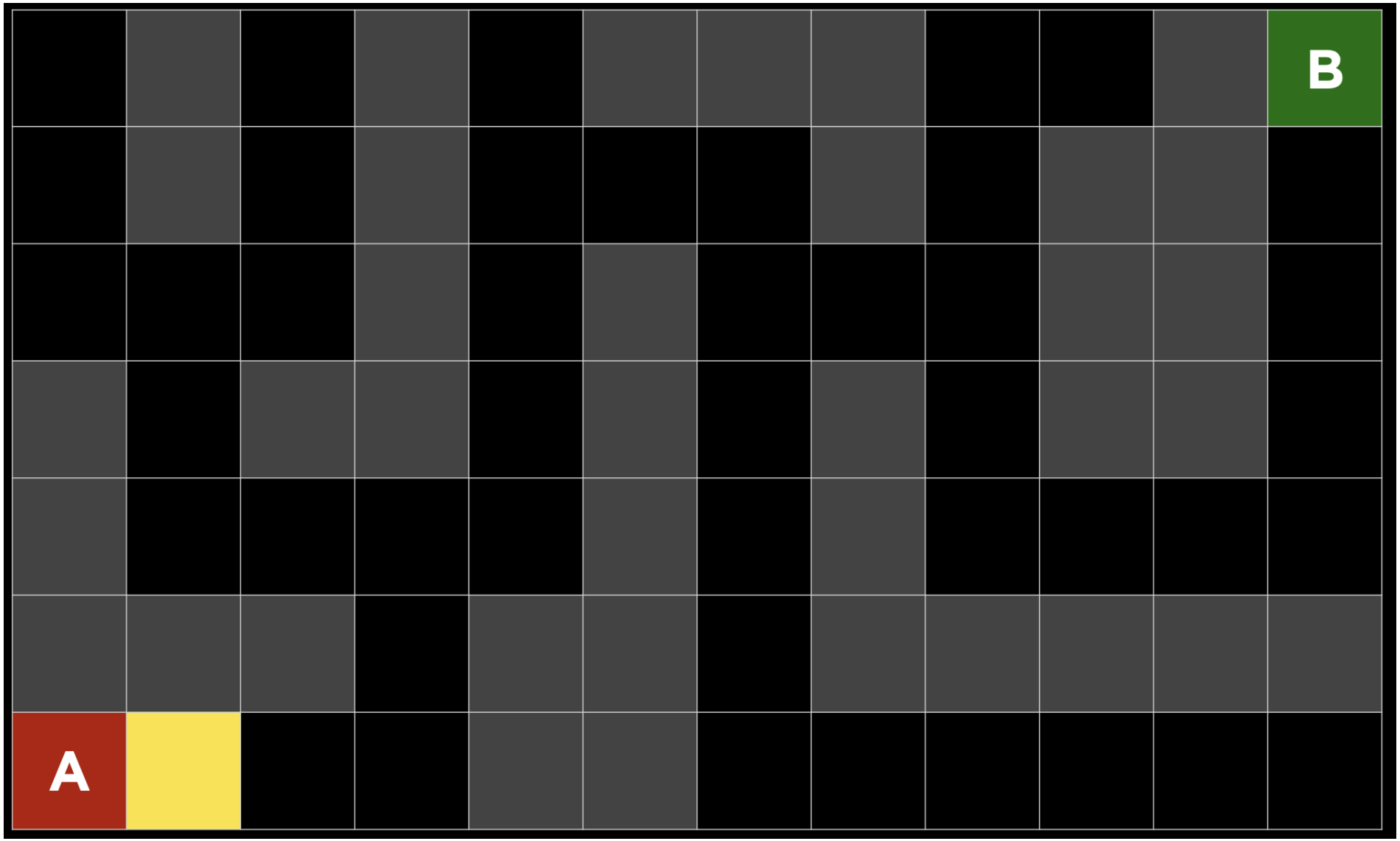
更多例子

https://visualgo.net/zh/dfsbfs

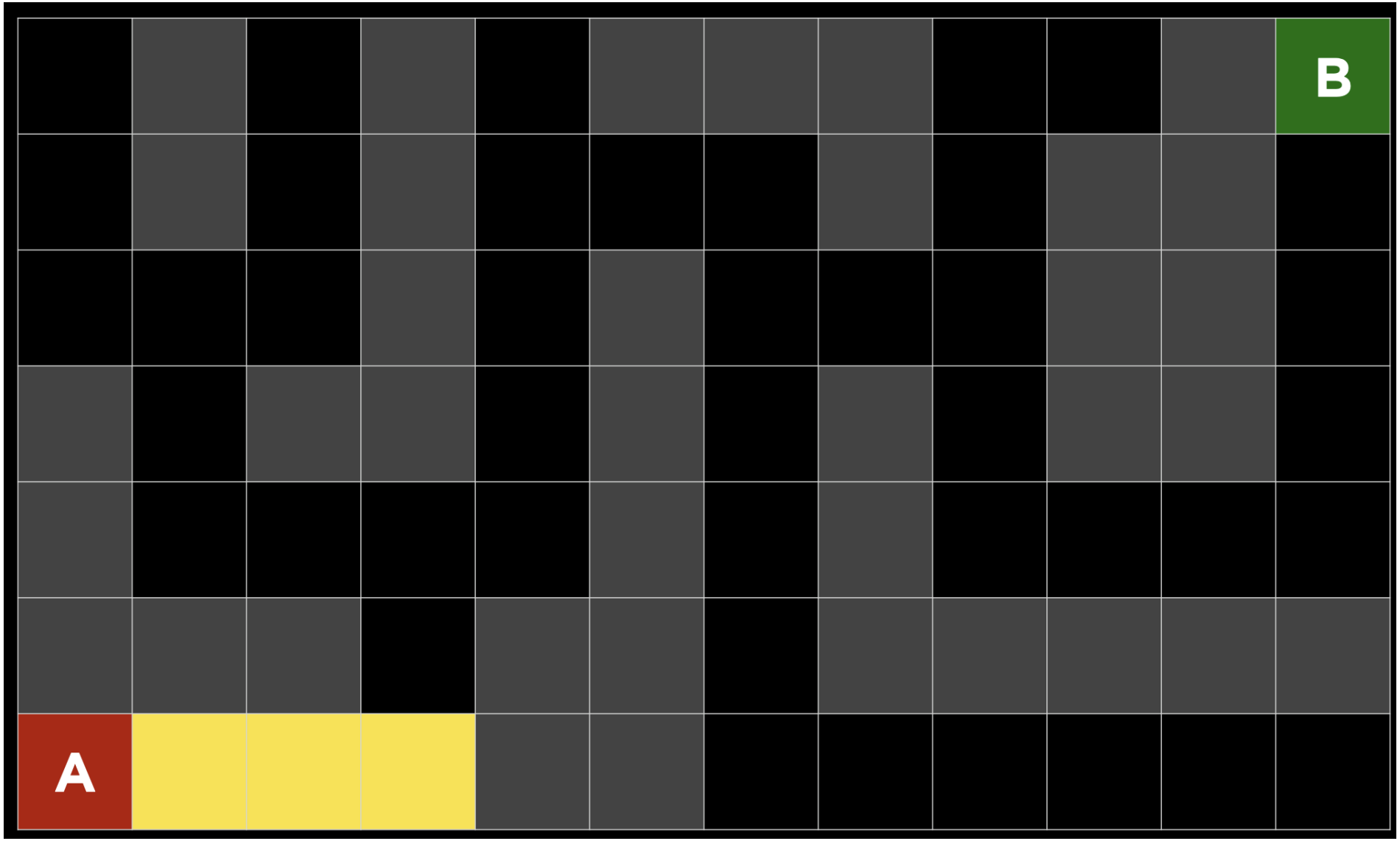
根据节点构建的顺序：深搜 vs. 广搜



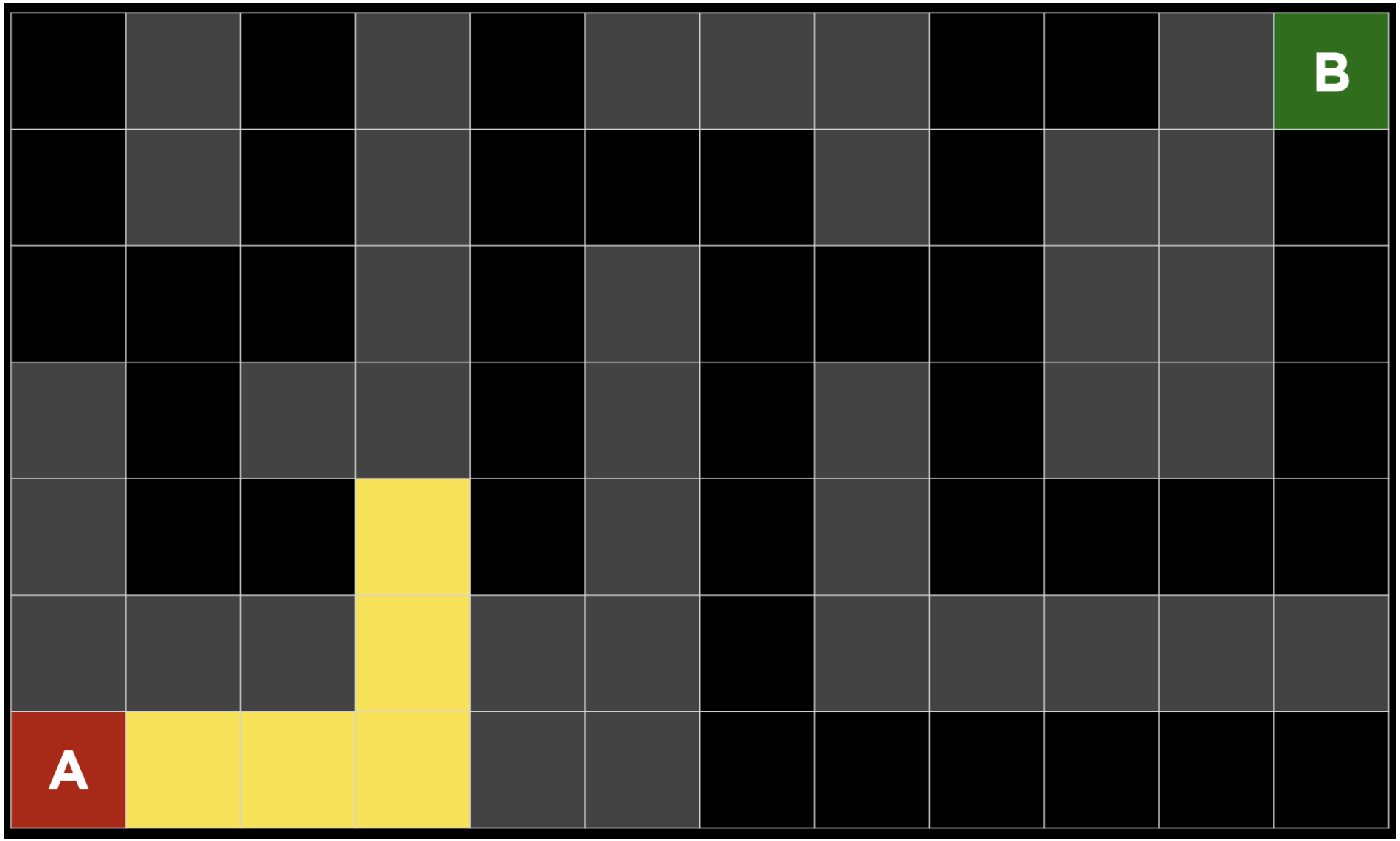
根据节点构建的顺序：深搜



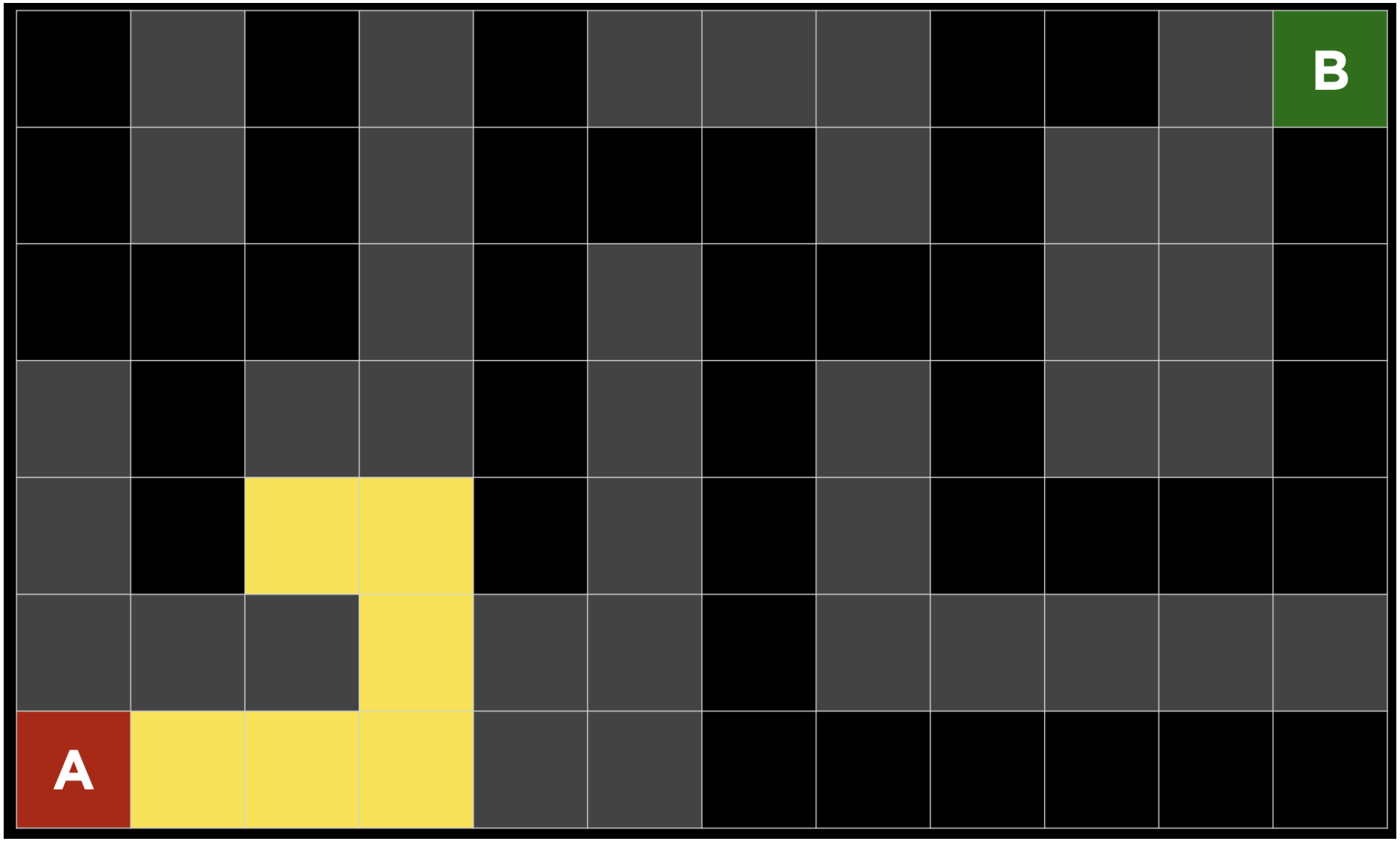
根据节点构建的顺序：深搜



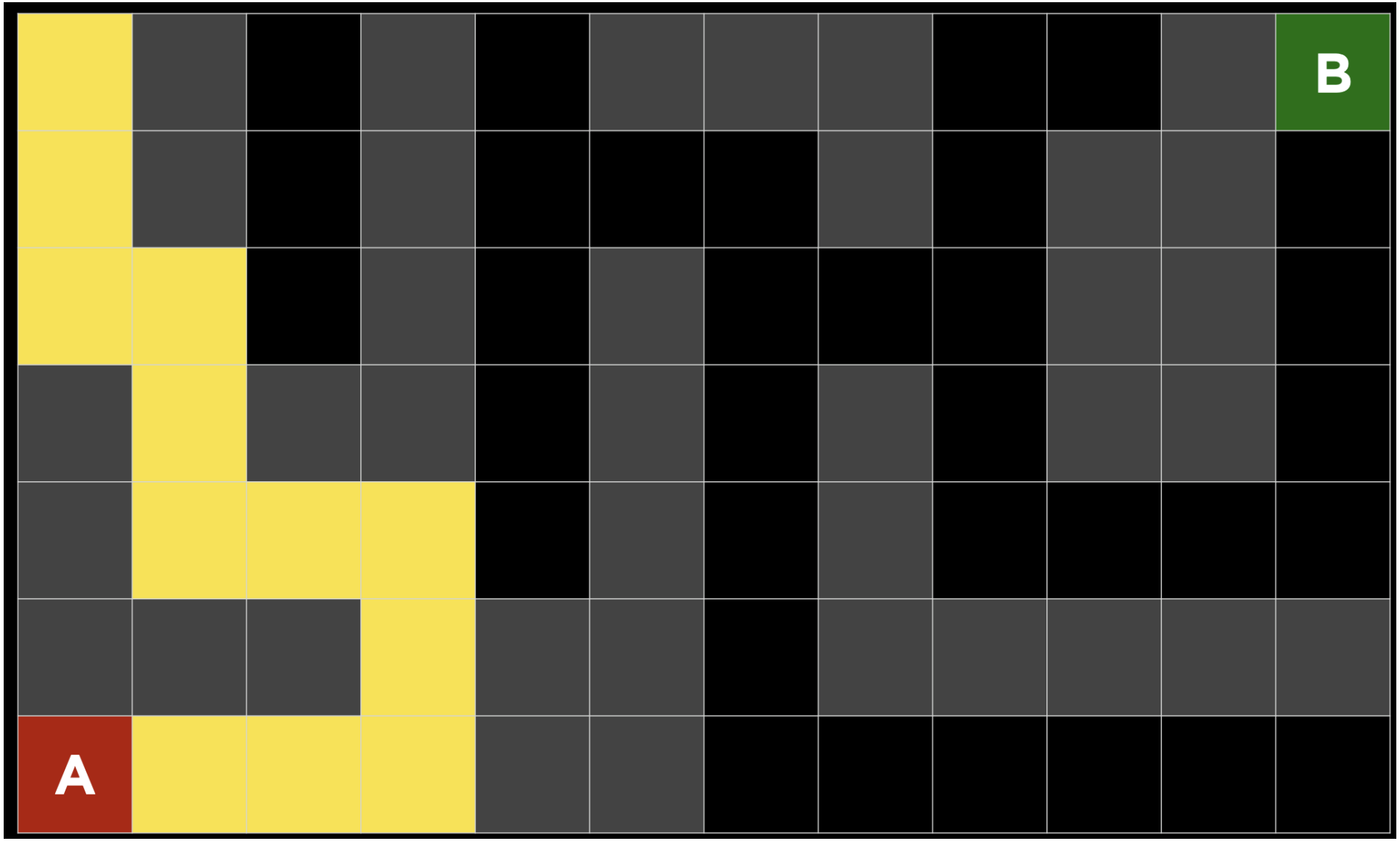
根据节点构建的顺序：深搜



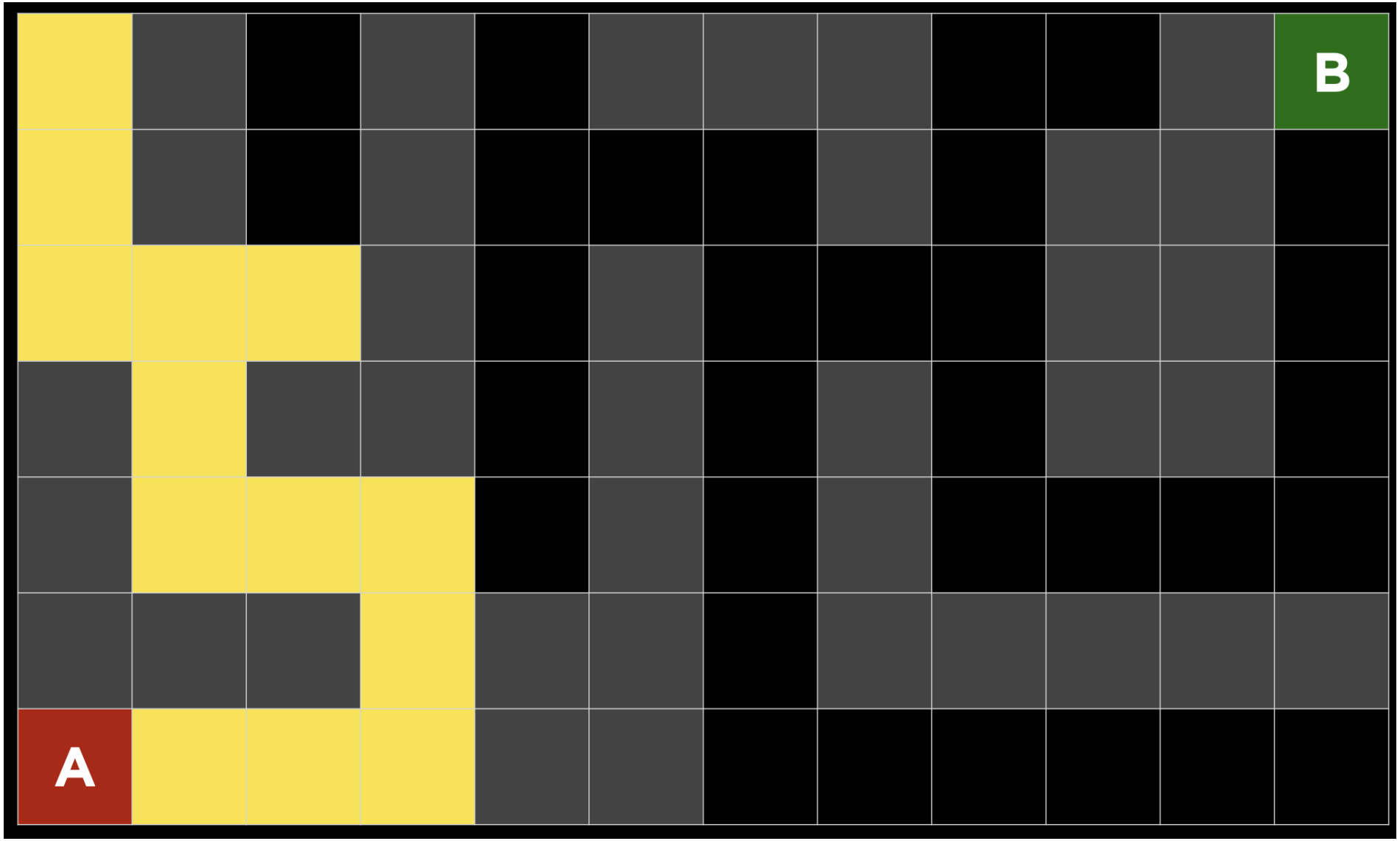
根据节点构建的顺序：深搜



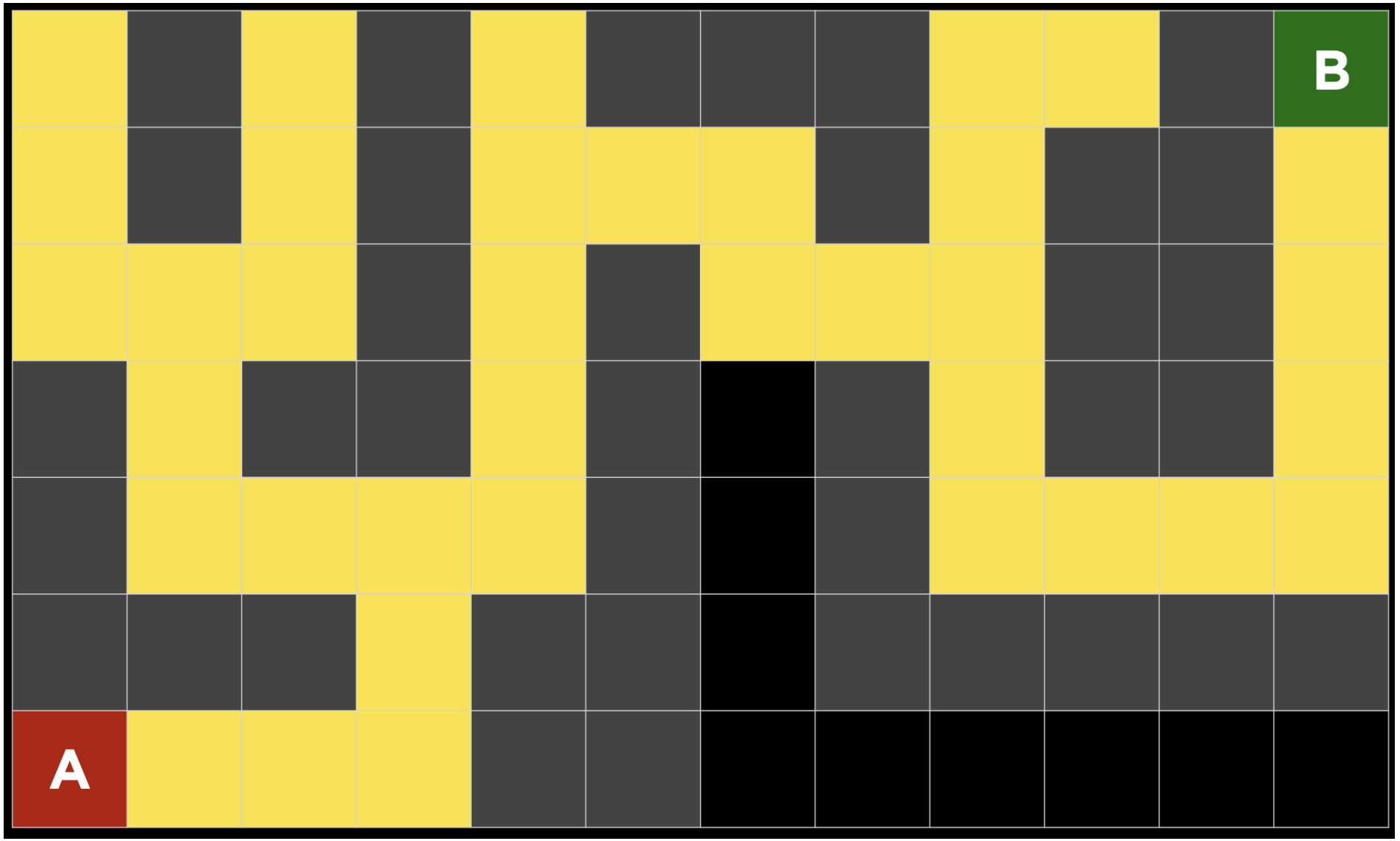
根据节点构建的顺序：深搜



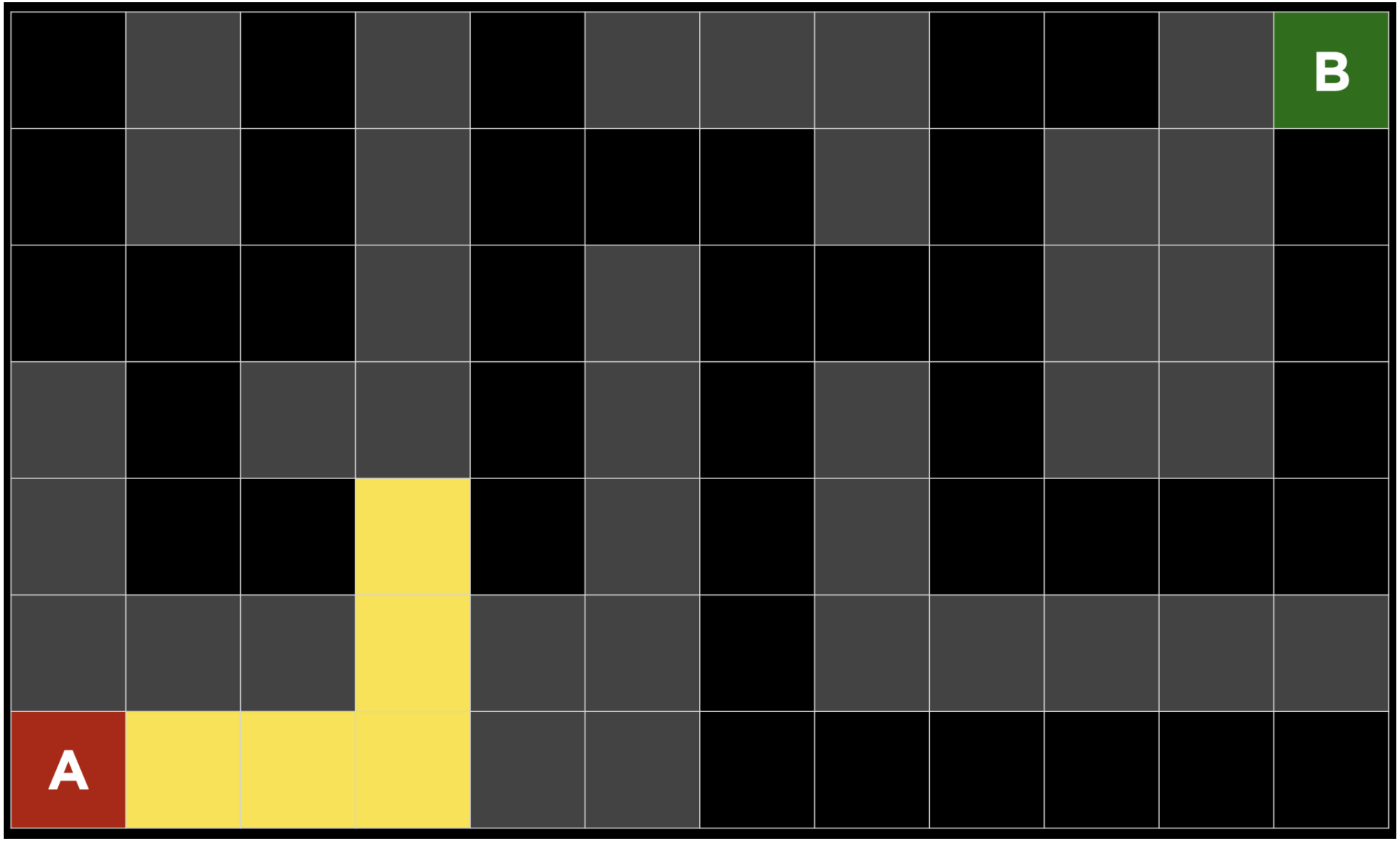
根据节点构建的顺序：深搜



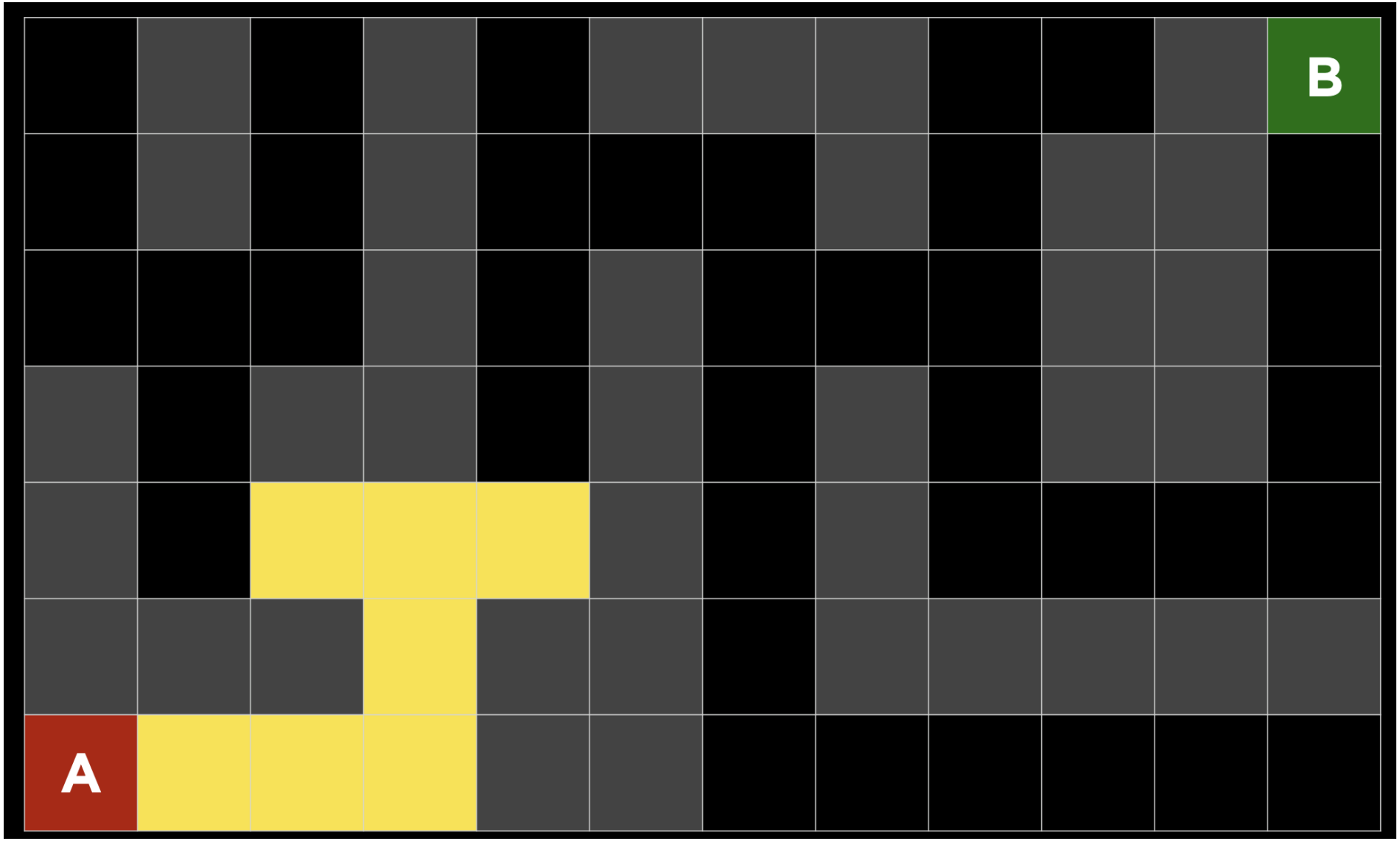
根据节点构建的顺序：深搜



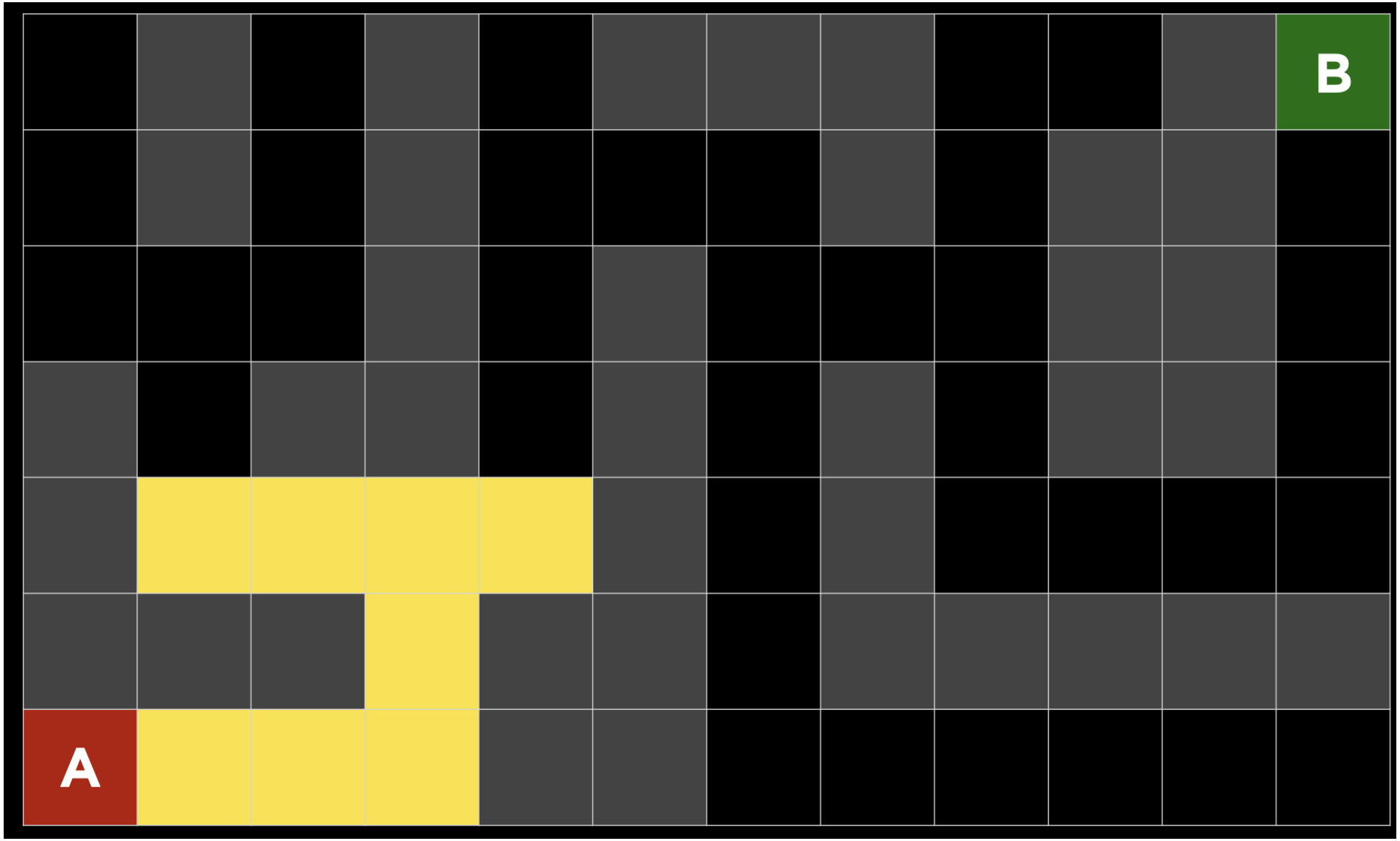
根据节点构建的顺序：广搜



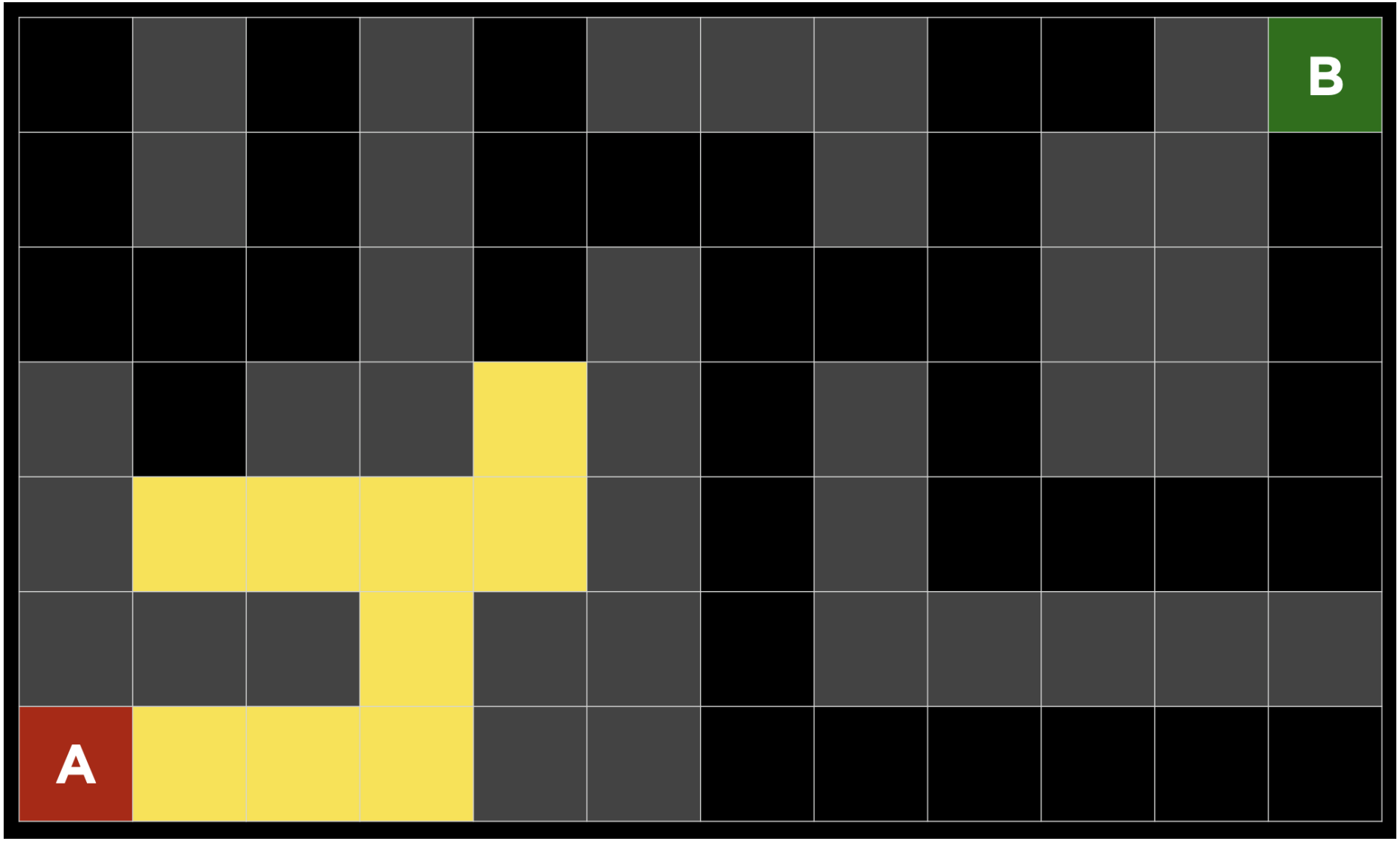
根据节点构建的顺序：广搜



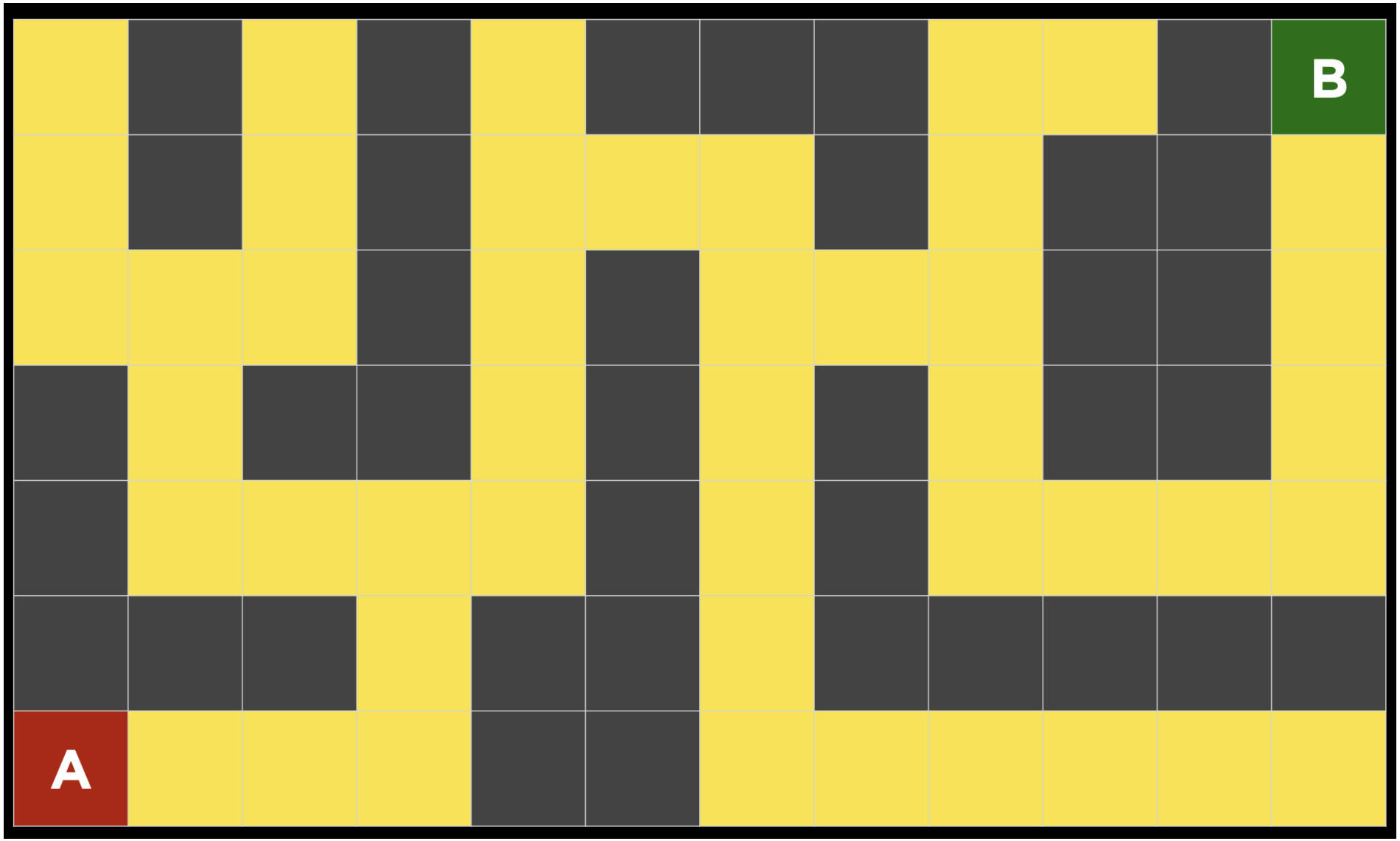
根据节点构建的顺序：广搜



根据节点构建的顺序：广搜



根据节点构建的顺序：广搜



如何用python实现搜索

北京大学 PEKING UNIVERSITY

- 有什么数据结构可以帮助我们?
- 什么样的计算结构?

如何用python实现搜索

北京大学 PEKING UNIVERSITY

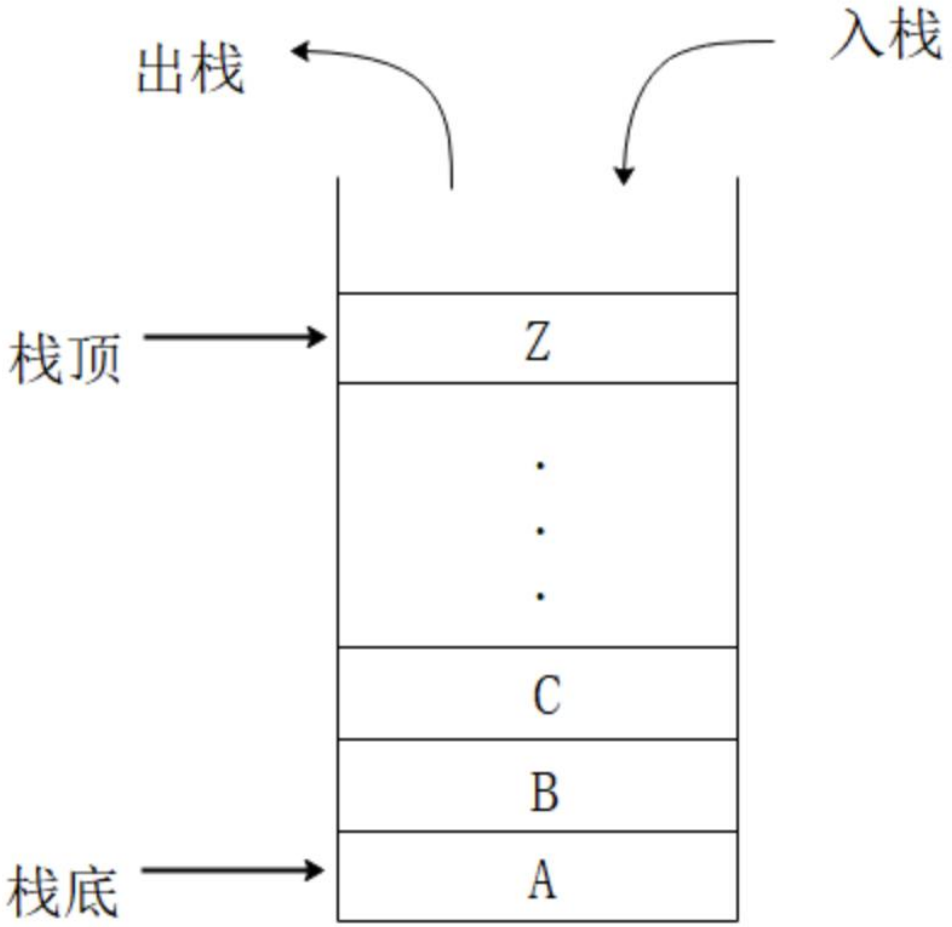
- 有什么数据结构可以帮助我们?

  stack （栈） vs. queue （队列）

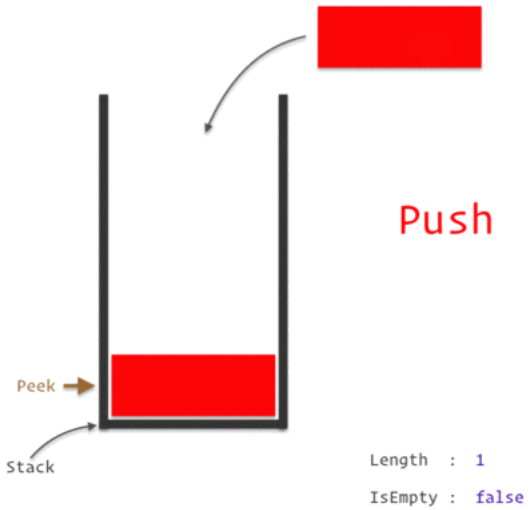
- 什么样的计算结构?

  recursion （递归） vs. loop （循环）

什么是栈（stack）



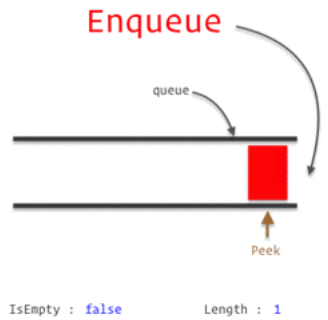
数据结构特点：后进先出



什么是队列（queue）

数据结构特点：先进先出





如何用python实现广搜

```
class SearchStack:
    def __init__(initial_states, actions):
        self.states = initial_states
        self.actions = actions
        self.current = None
        ……..
    def is_empty():
    def goal_reached():
    def current_state():
    def expand():
```

如何用python实现广搜

```
class SearchStack:
    def current_state():
        self.current = self.states.popleft()
        return self.current      //我们从头开始考虑，但是python自带的list支持popleft吗?
    def expand():
        for action in self.actions:
            self.states.append(self.transition(self.current,action)) //怎么判重，怎么判断哪些action是可以的?
```

如何用python实现广搜

```
def search(initial_states, goal_states,.....):
    search_stack = SearchStack(....)
    while not search_stack.is_empty():
        current = search_stack.current()
        search_stack.expand()
        if search_stack.goal_reached():
            return True
    return False
```

搜索概览小结

北京大学 PEKING UNIVERSITY

- 什么是搜索?
- 一个搜索问题或者说搜索算法应该需要包含什么必要的部分
- 暴力搜索（深搜和广搜）

预告

- 为什么我们平等对待每一个动作？ -> 启发搜索
- 当有一个对手和我们博弈的时候？ -> minimax和剪枝
- 如何系统性的用符号表达状态？ -> 逻辑
- 如果环境的转变是不确定性的？ -> 不确定性和概率推理
- 搜索空间太大了？ -> 蒙特卡洛搜索

谢谢

北京大学
PEKING UNIVERSITY

