



# 程序设计实践（一）

哈尔滨工业大学 计算机学院  
任课教师：孙大烈教授  
助教：付万增



# 搜索基础

- 一、深度优先搜索
- 二、广度优先搜索
- 三、实践出真知

## 背景知识

无人驾驶飞机简称“无人机”，英文缩写为“UAV”，是利用无线电遥控设备和自备的程序控制装置操纵的不载人飞机。





我们本节课的任务：制作无人机。

？ ？ ？

**NONONO!**

学习目标：设计无人机寻路算法。



## 核心问题

# 如何控制计算机自动走迷宫？





实现算法：搜索

搜索算法是利用计算机的高性能来有目的地穷举一个问题的部分或所有的可能情况，从而求出问题的解的一种方法。

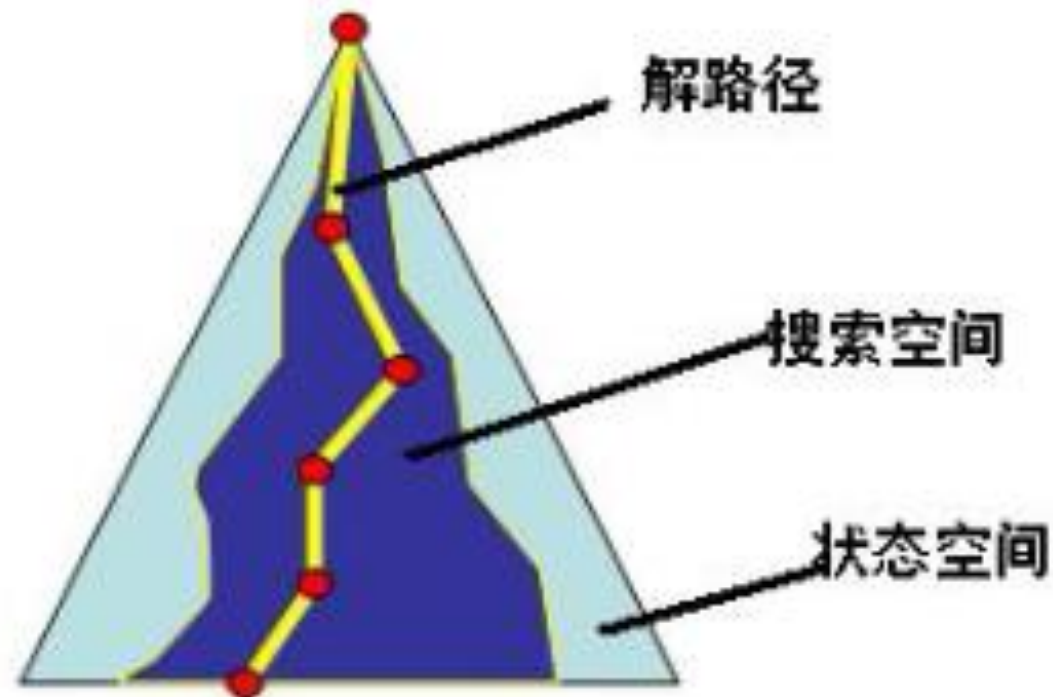
有一定的方向性和目标性

本质：穷举所有可能





搜索算法是在解的空间里，从一个状态转移（按照要求拓展）到其他状态，这样进行下去，将解的空间中的状态遍历，找到答案（目标的状态）。







状态(**state**)是对问题在某一时刻进展情况的数学描述，或者是数学抽象。

每一个状态都会是答案的一个“可能的”解。状态的转移就是问题从一个状态转移到另一个状态，这样就可以进行搜索的一步步延伸，最后要得到的解也是其中的一个状态。





进行状态转移的算法

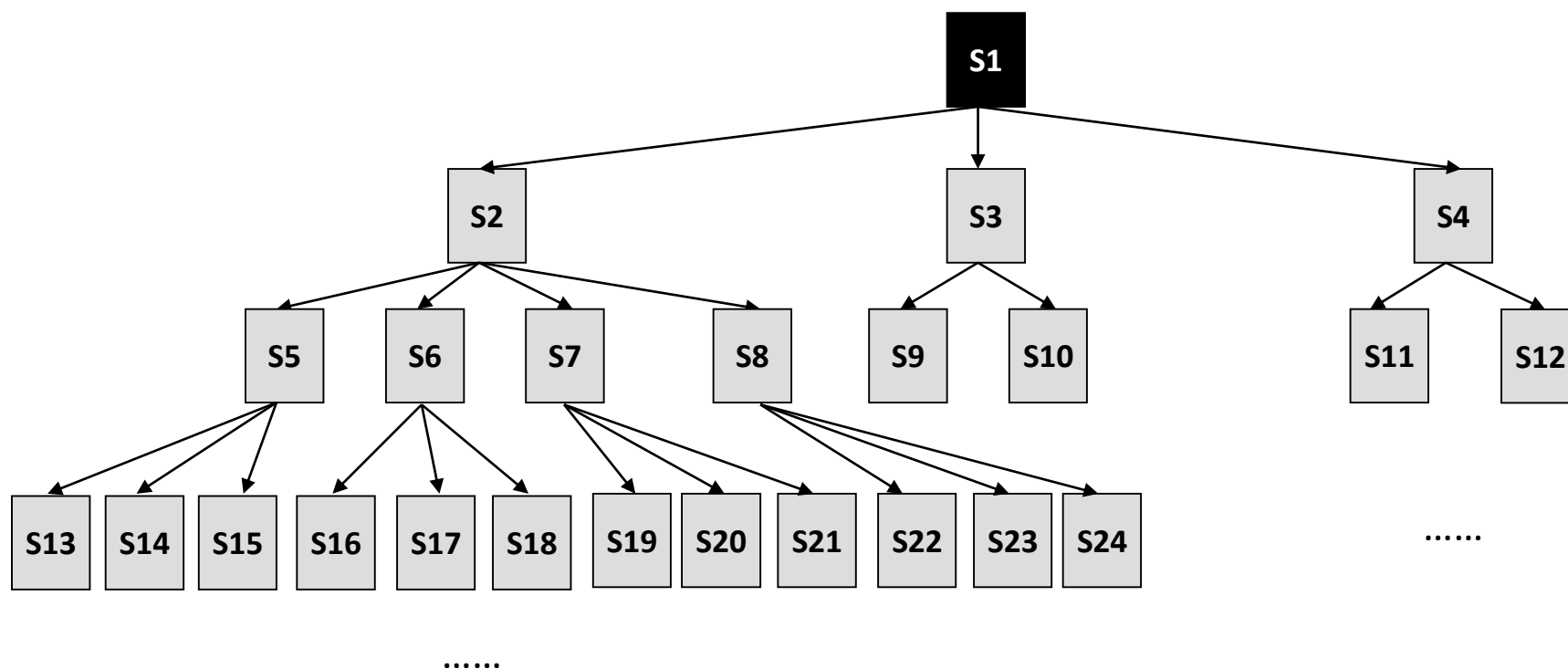
深度优先搜索（DFS）

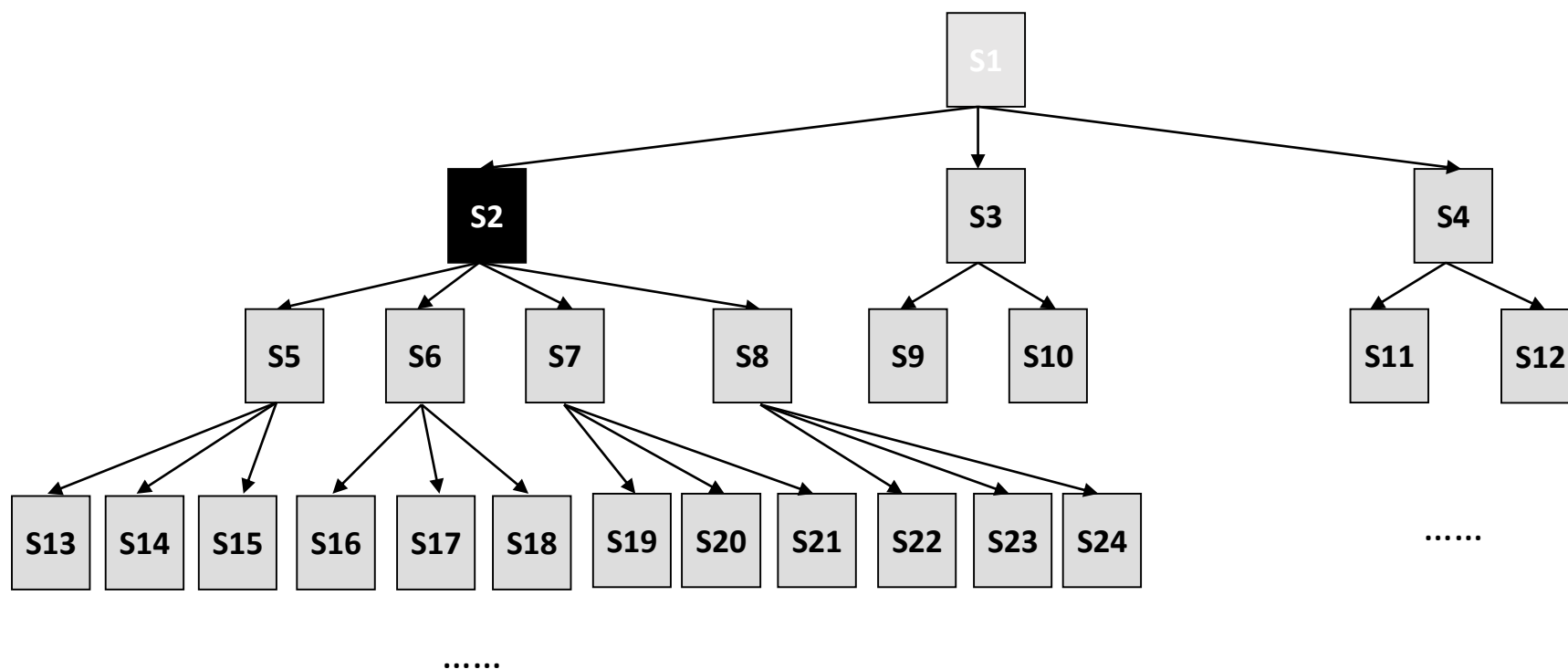
广度优先搜索（BFS）

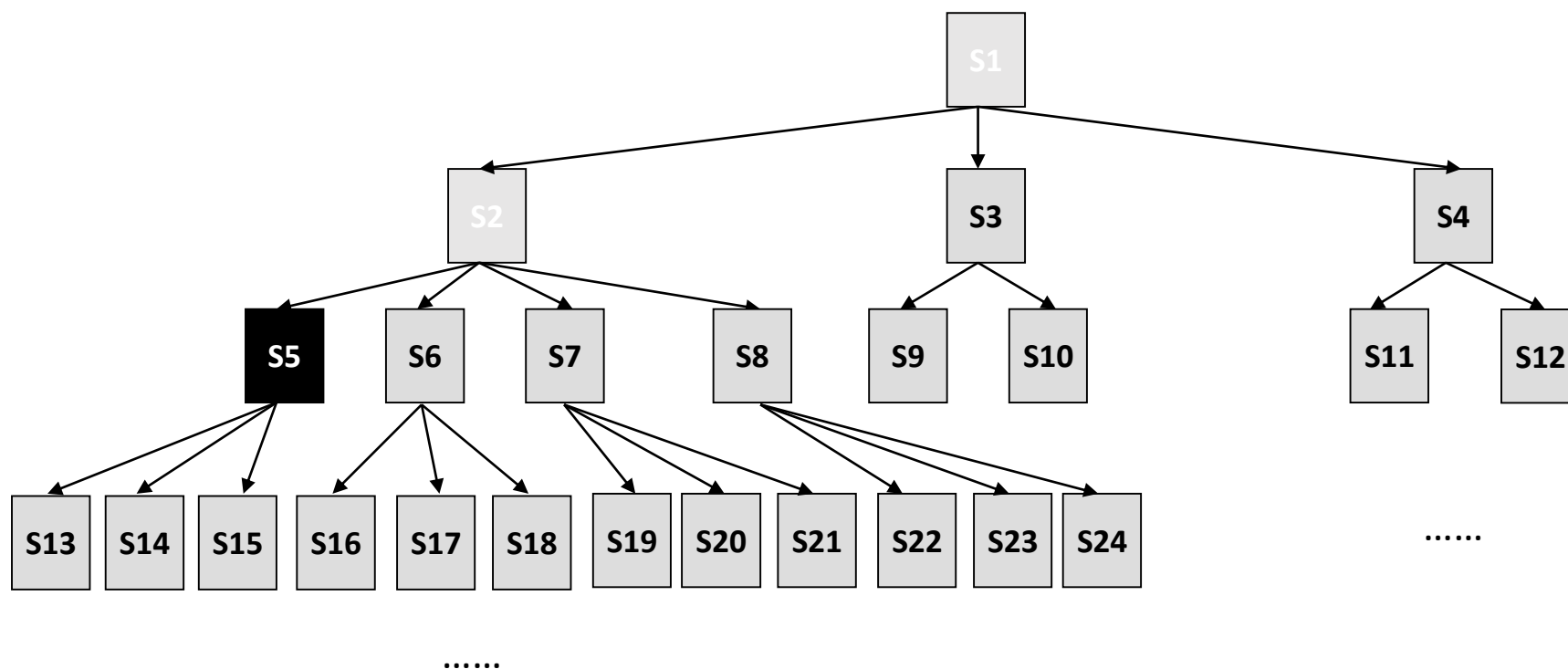


基本思想：

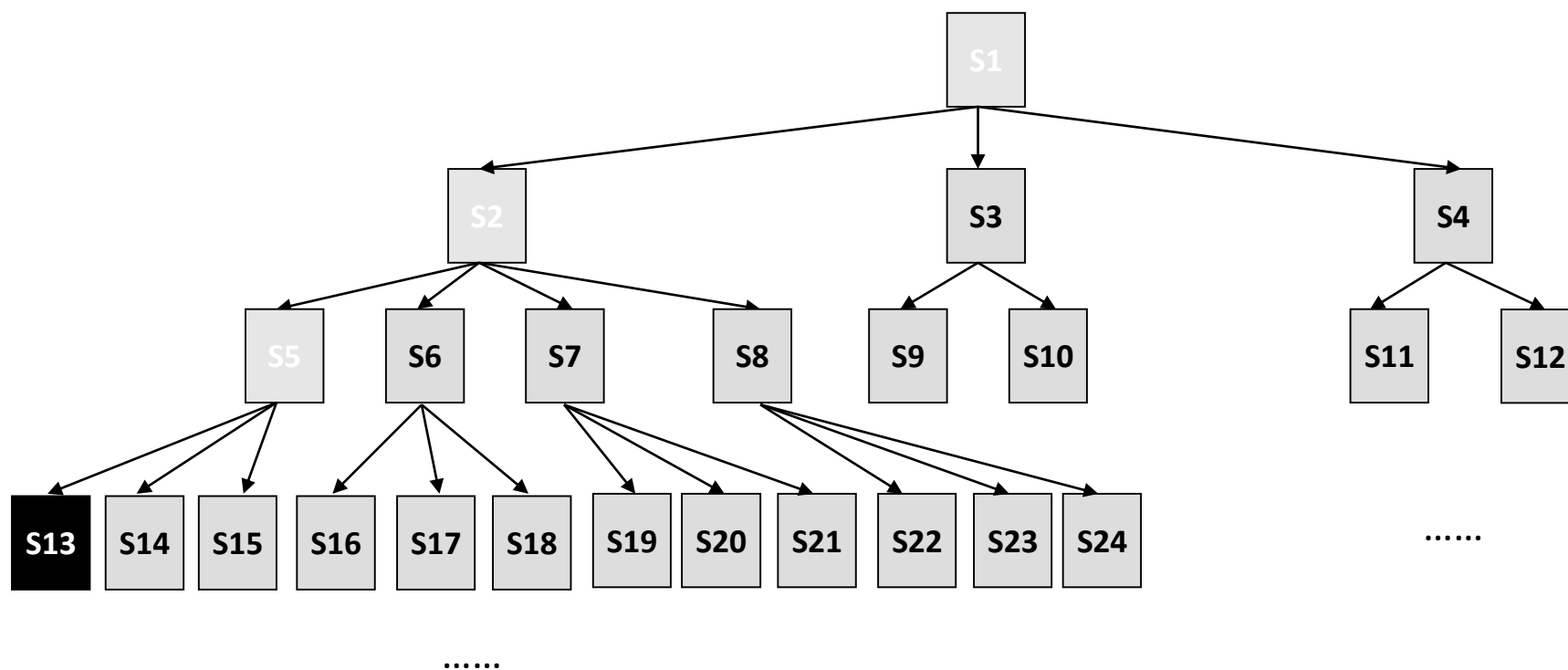
- 1.从初始状态，生成**搜索树**下一层任一个结点
- 2.检查是否出现目标状态
- 3.若未出现，以此状态利用规则生成搜索树下一层任一个结点，再检查，重复过程一直到叶节点（即不能再生成新状态节点）
- 4.当它仍不是目标状态时，回溯到上一层结果，取另一可能扩展搜索的分支。
- 5.采用相同办法一直进行下去，直到找到目标状态为止。

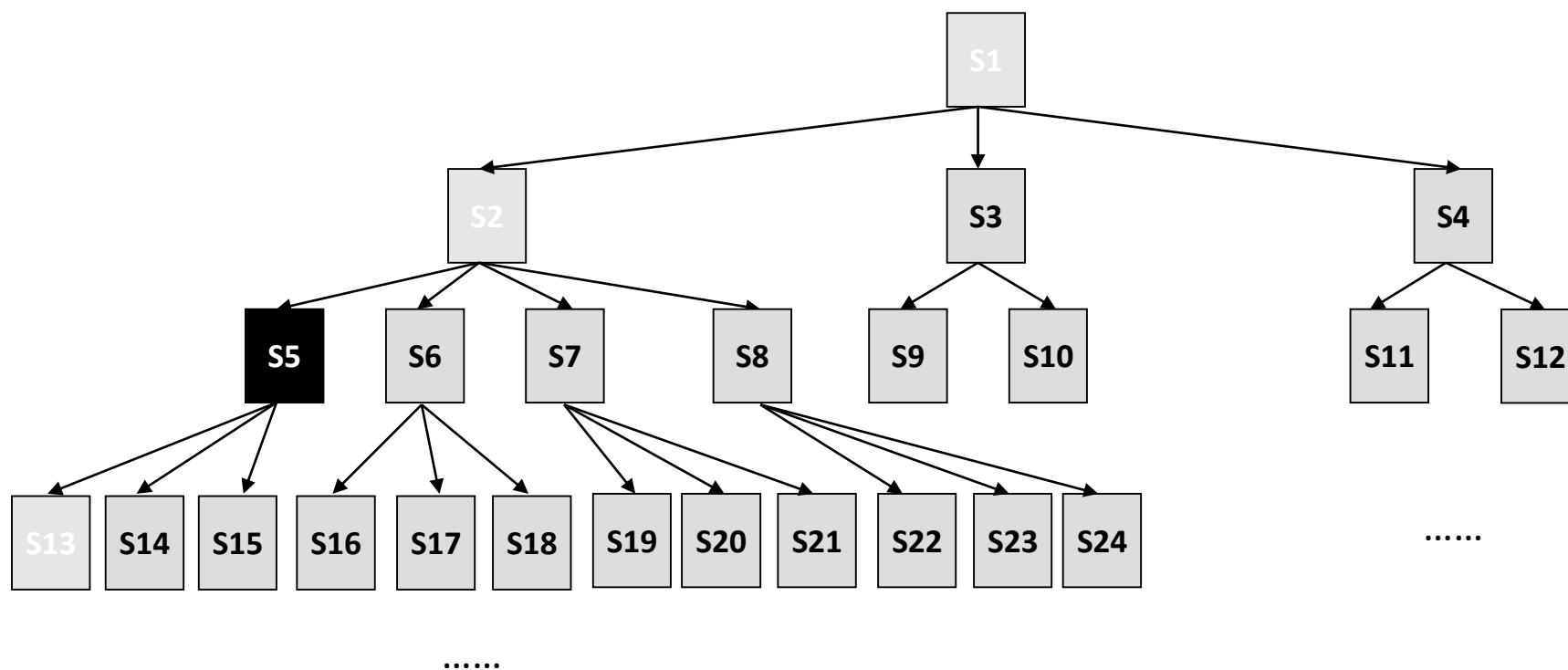


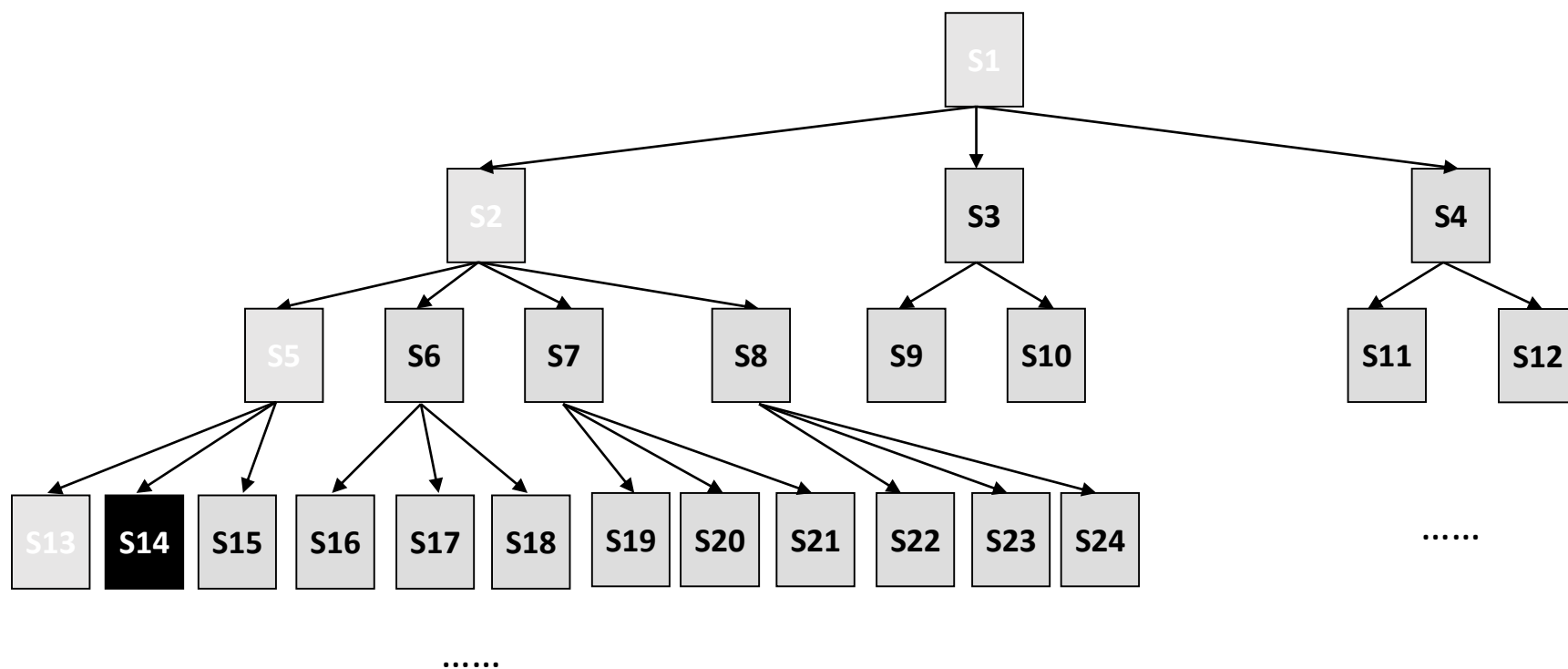


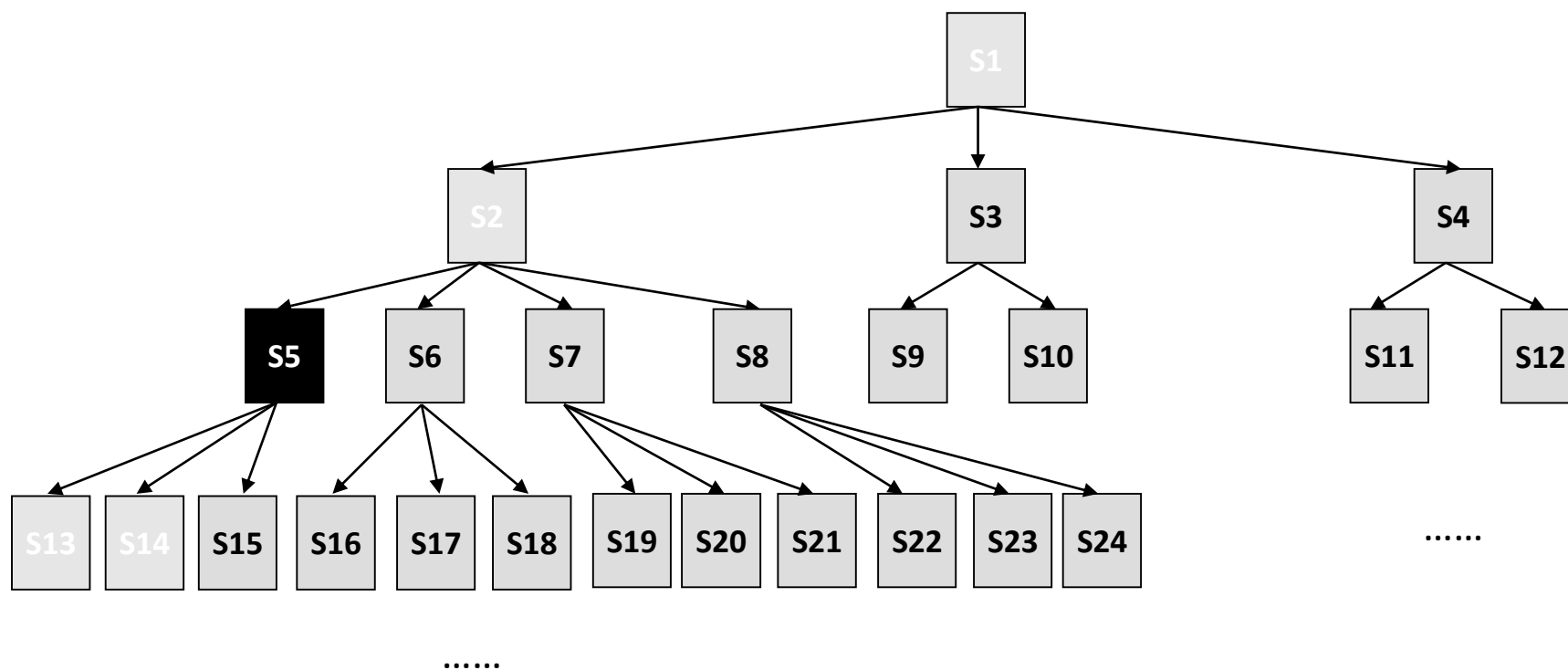


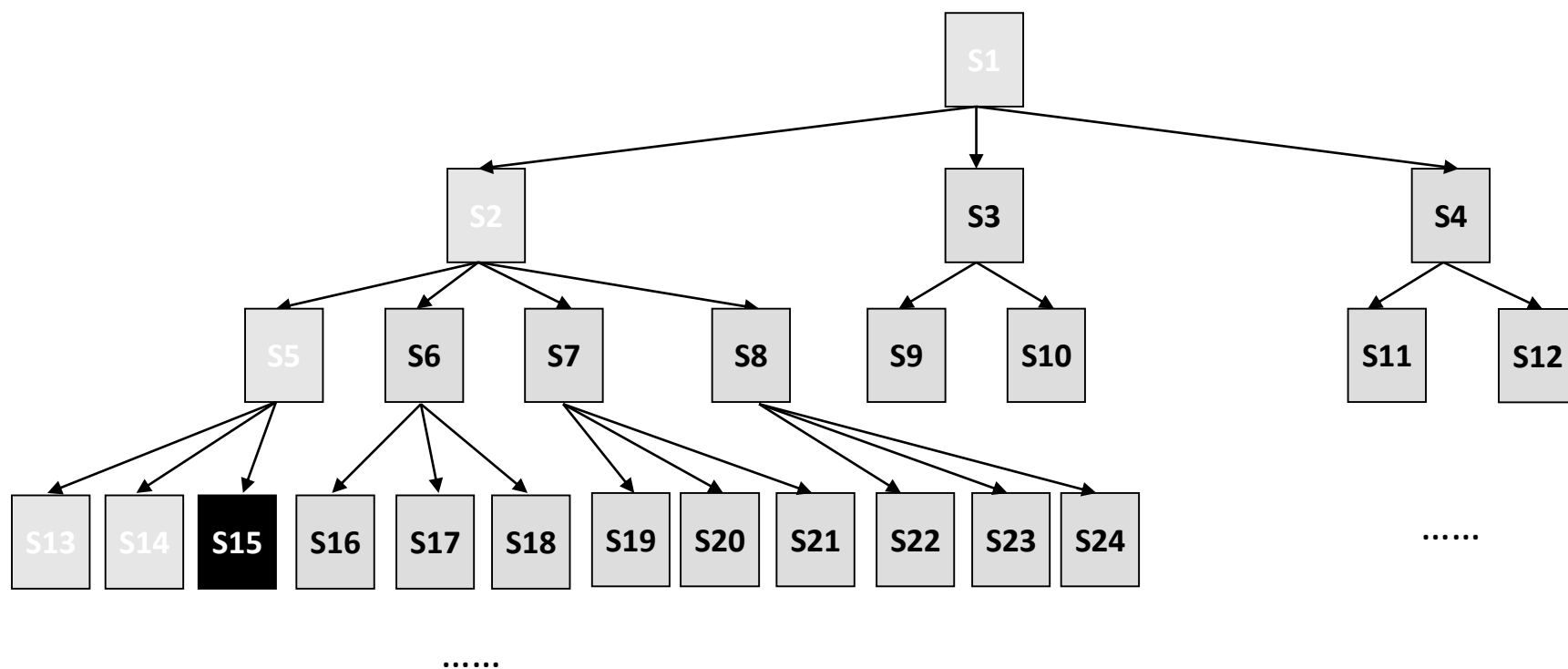




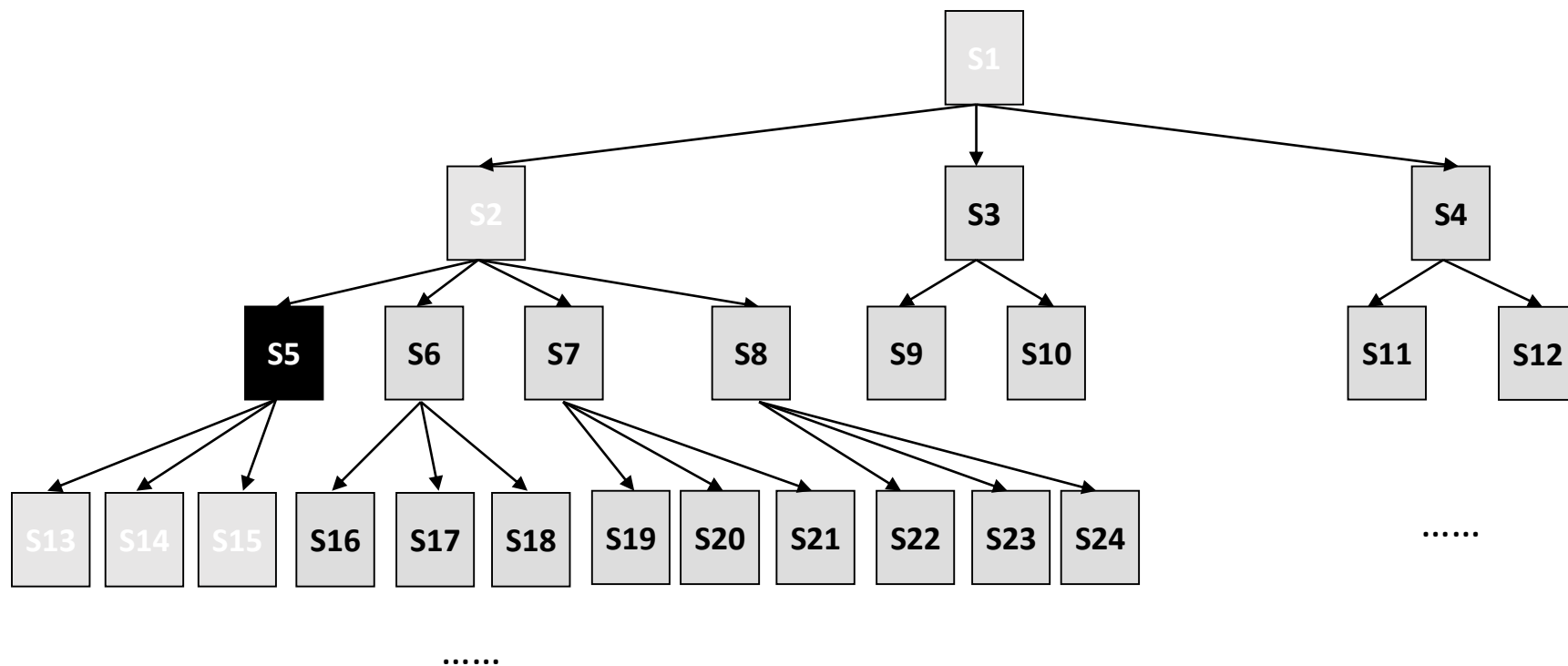


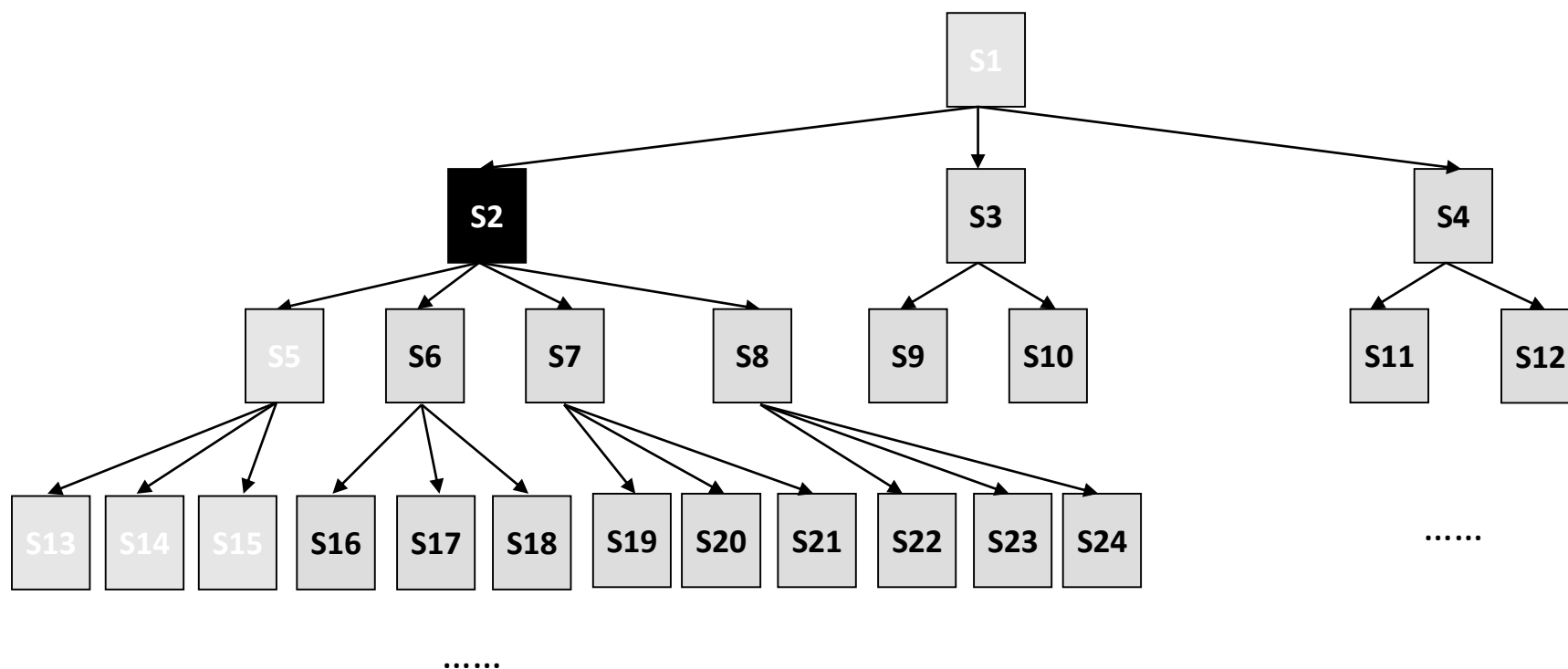


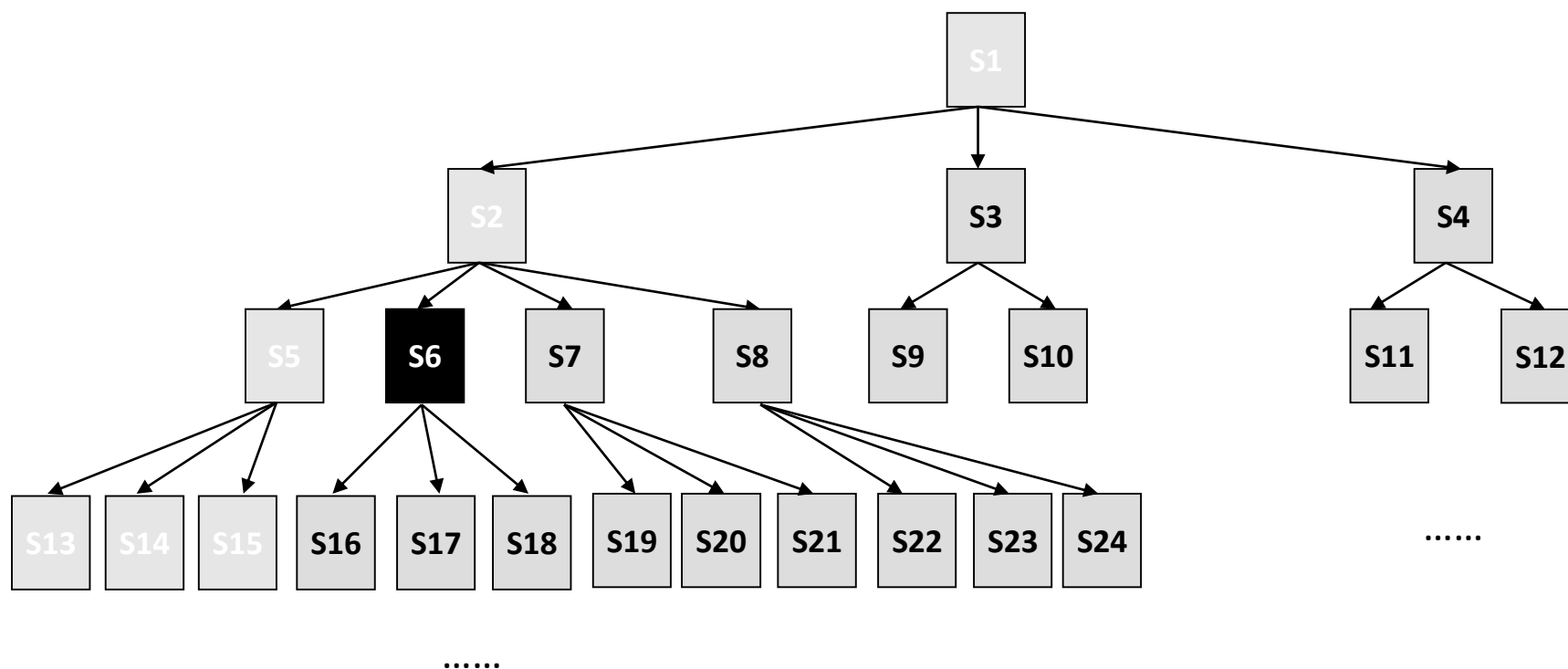


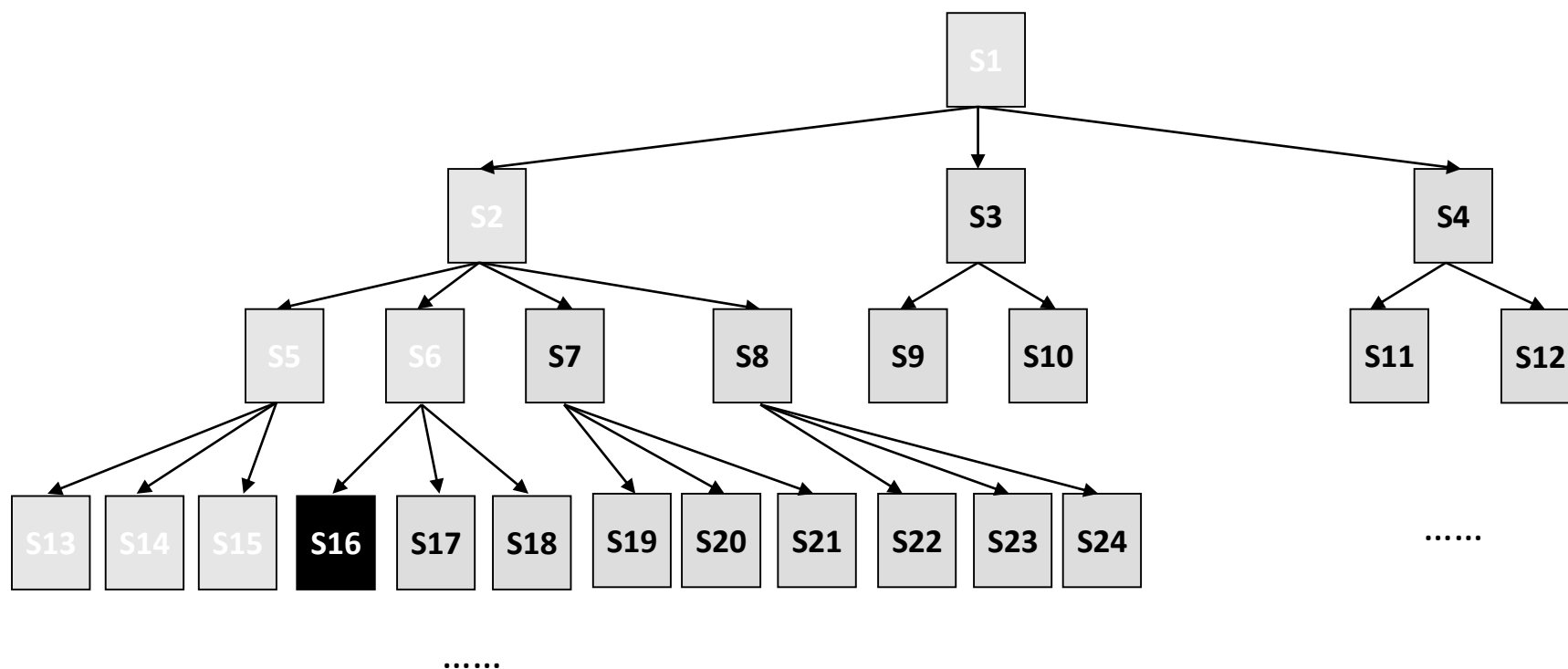














黑色填充代表当前探索的状态，白色数字为已经搜索过的状态，黑色数字为未搜索的状态。在过程中遇到搜索的解则退出，找到答案。

特点：以深度优先为前提进行扩展。

这就是DFS的过程，状态必须在遍历完所有它的子状态之后，才能继续进行对同一层中下一个状态的遍历。

符合栈的先进后出（FILO）的性质。

与迷宫问题的关系：将迷宫路径抽象成状态集合，每一步都是一种状态。枚举所有状态即枚举所有可行路径。找到目标状态即找到迷宫出口。





利用栈实现？

具体实现过程

- 1 每次取出栈顶元素，对其进行拓展。
- 2 若栈顶元素无法继续拓展，则将其从栈中弹出。继续1过程。
- 3 不断重复直到获得目标状态（取得可行解）或栈为空（无解）。

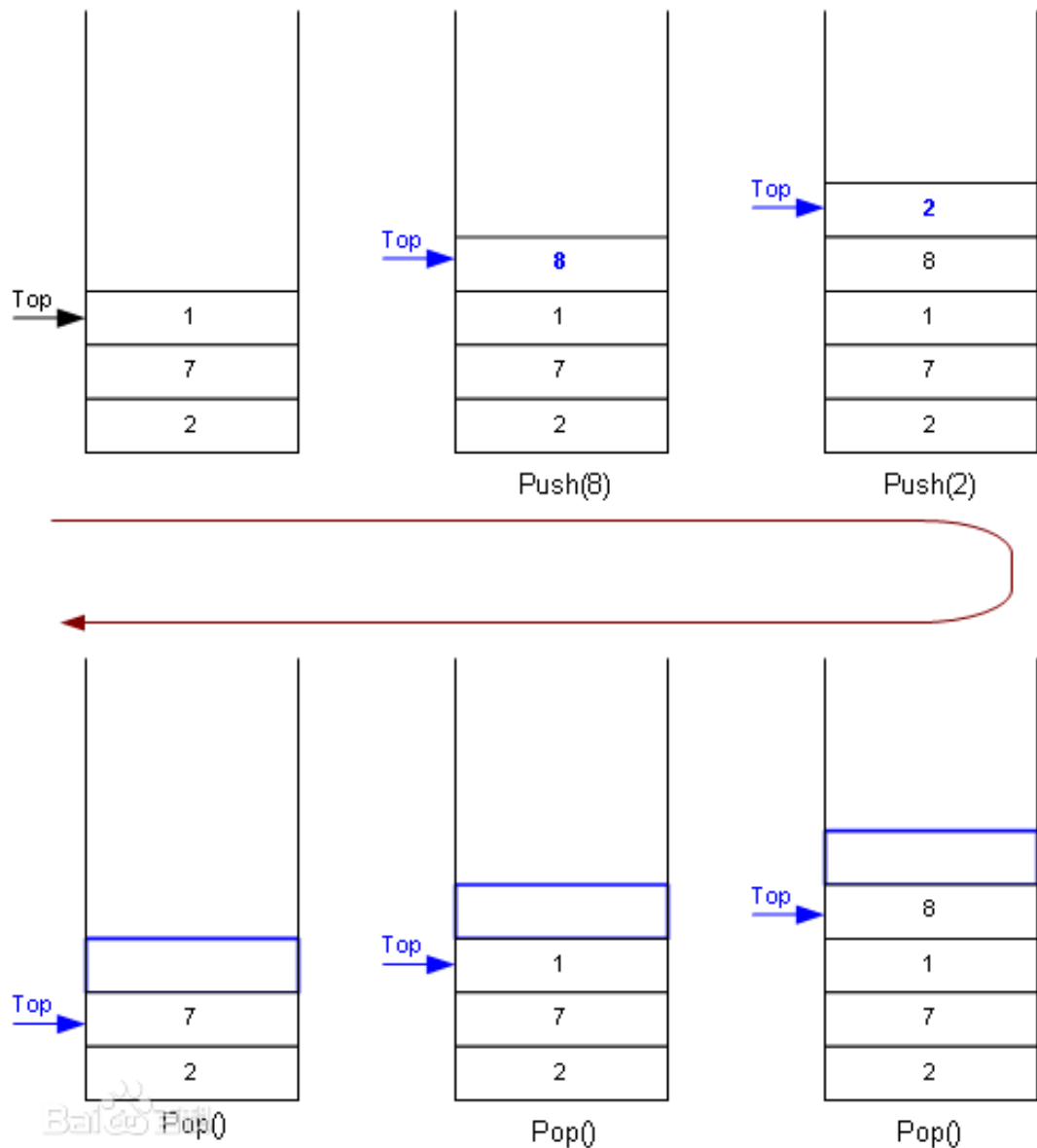
更好的办法：递归？

回忆栈和递归，想一想两者的联系。

# 栈



- 栈(stack)
- 栈顶: top
- 入栈 push()
- 出栈 pop()

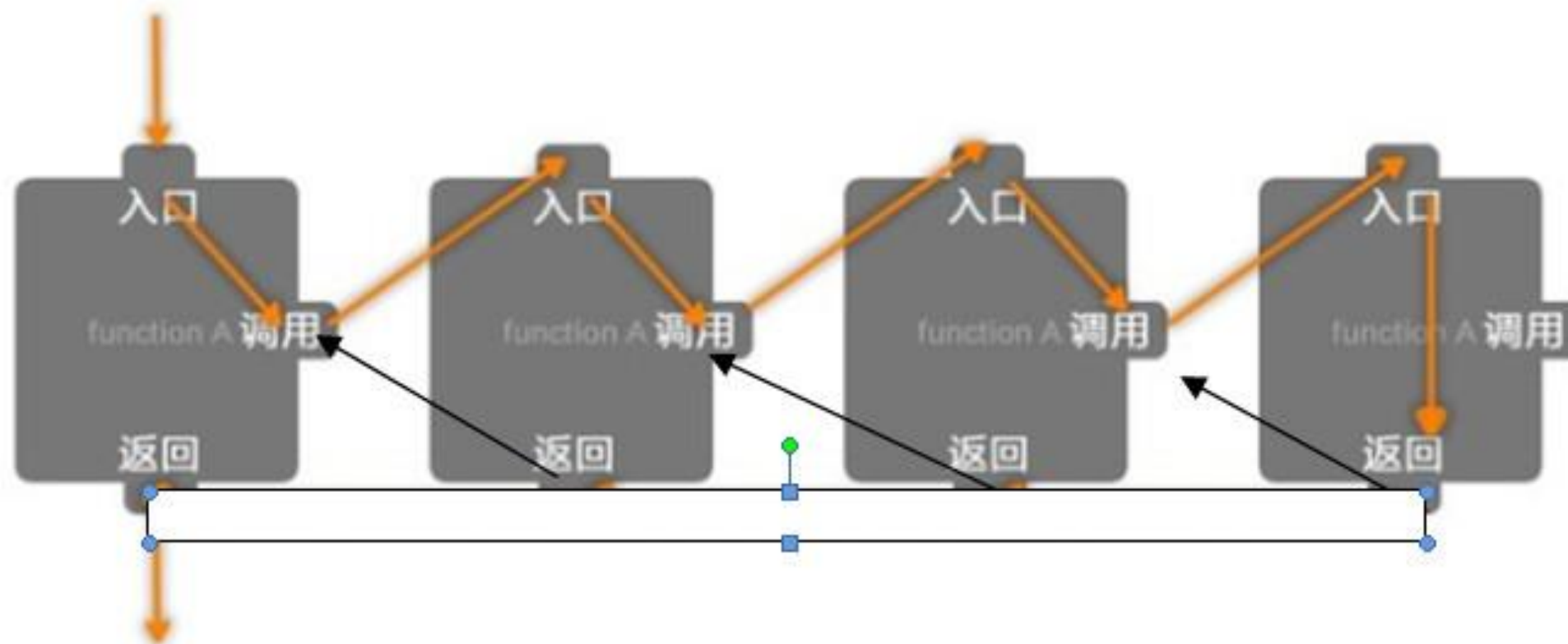


Stack的Push和Pop, 遵循先进后出规则

# 递归



递归过程中发生了什么？





更好的办法：递归？

回忆栈和递归，想一想两者的联系。

递归是什么？

子程序调用本身

如何实现？

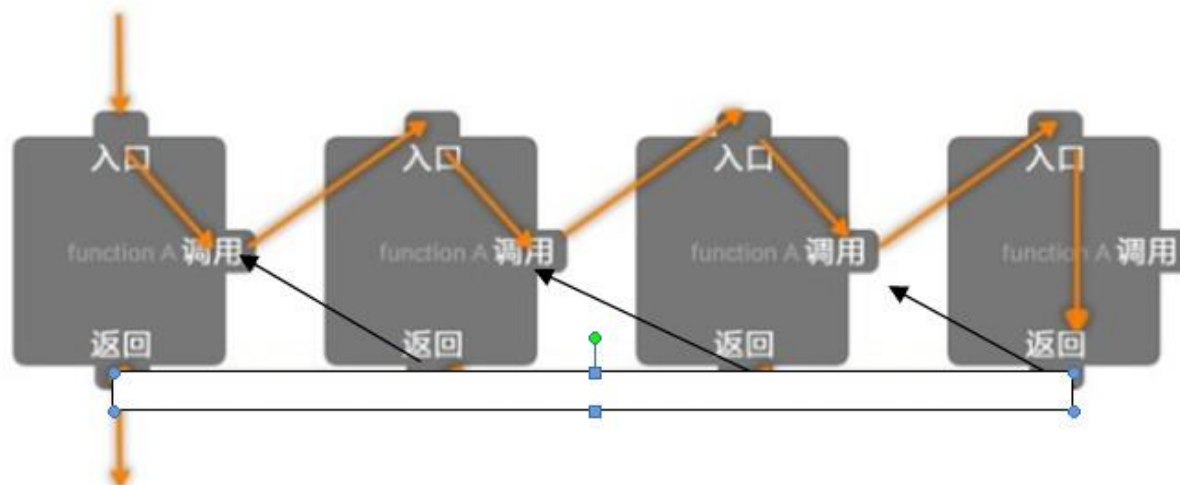
利用栈结构实现！！！！

当发生递归调用时，

将现有程序压入系统堆栈，

执行子程序，继续调用继续压栈，直到达到终止条件。

然后反复弹栈，直到返回主函数，继续顺序执行。





**A:**工大毕业的土木专业+建筑专业+包工头带的一队人，能自己设计房子盖出来不？

**B:**有包工头就行了。

**A:**？ ？ ？

**B:**农村那种房子，包工头都熟络于心了





**具体的迷宫问题：**迷宫大小 $n*m$ ，用字符串表示迷宫。其中，#表示墙，\*表示路径，S表示起点，T表示终点。

**Input**

5 6

###T#\*

S#\*\*\*\*\*

\*##\*##

\*\*\*\*\*

\*#\*#\*#

**Output**

(1,0)->(2,0)->(3,0)->(3,1)->(3,2)-  
>(3,3)->(2,3)->(1,3)->(0,3)

###T#\*

S#\***\***\*\*\*

**\***##**\***##

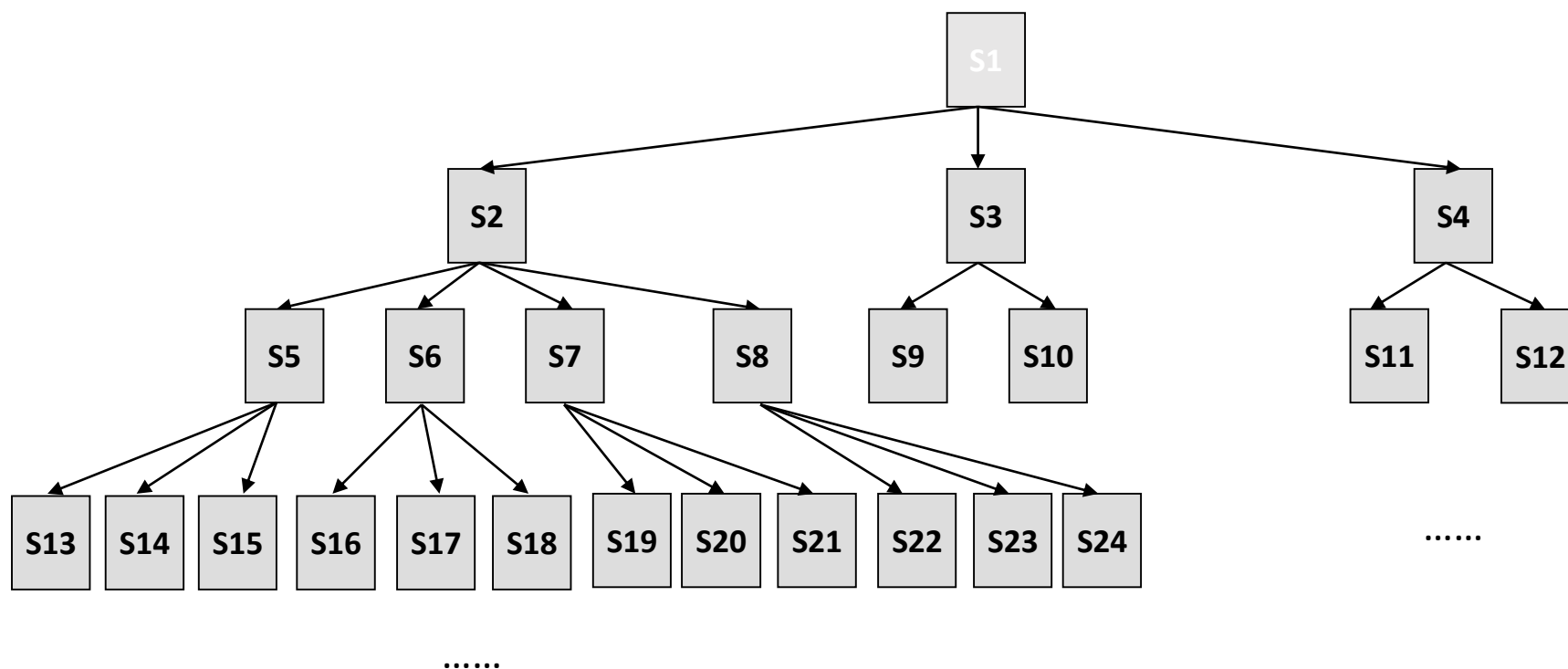
**\*****\*****\*****\*****\*****\***

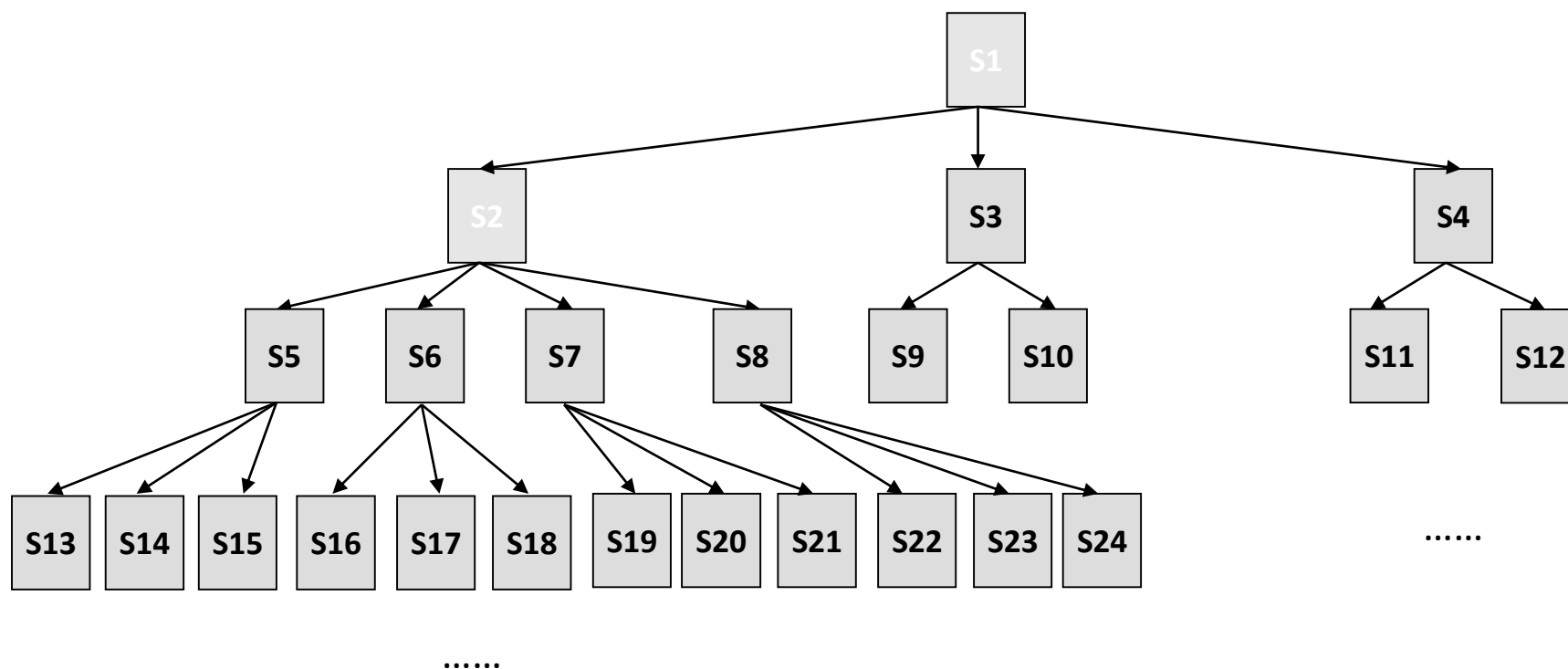
\*#\*#\*#

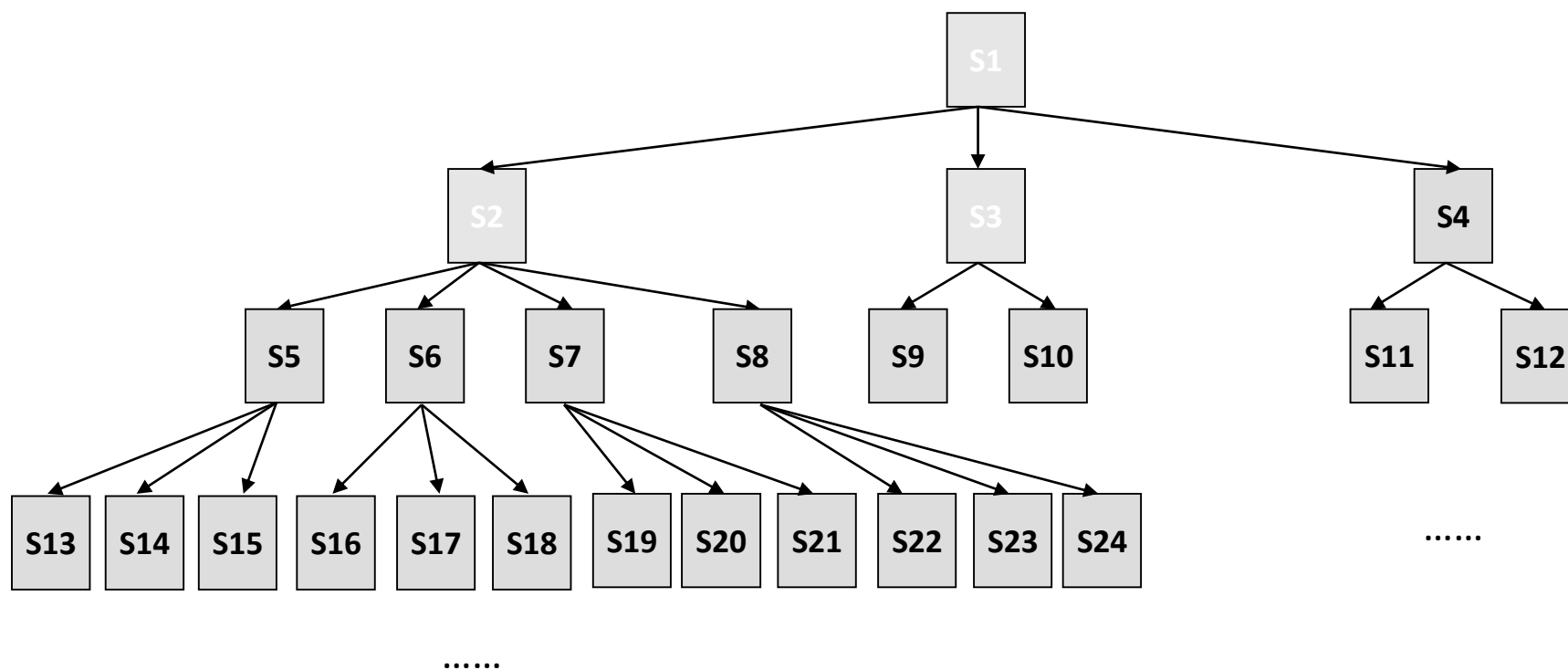


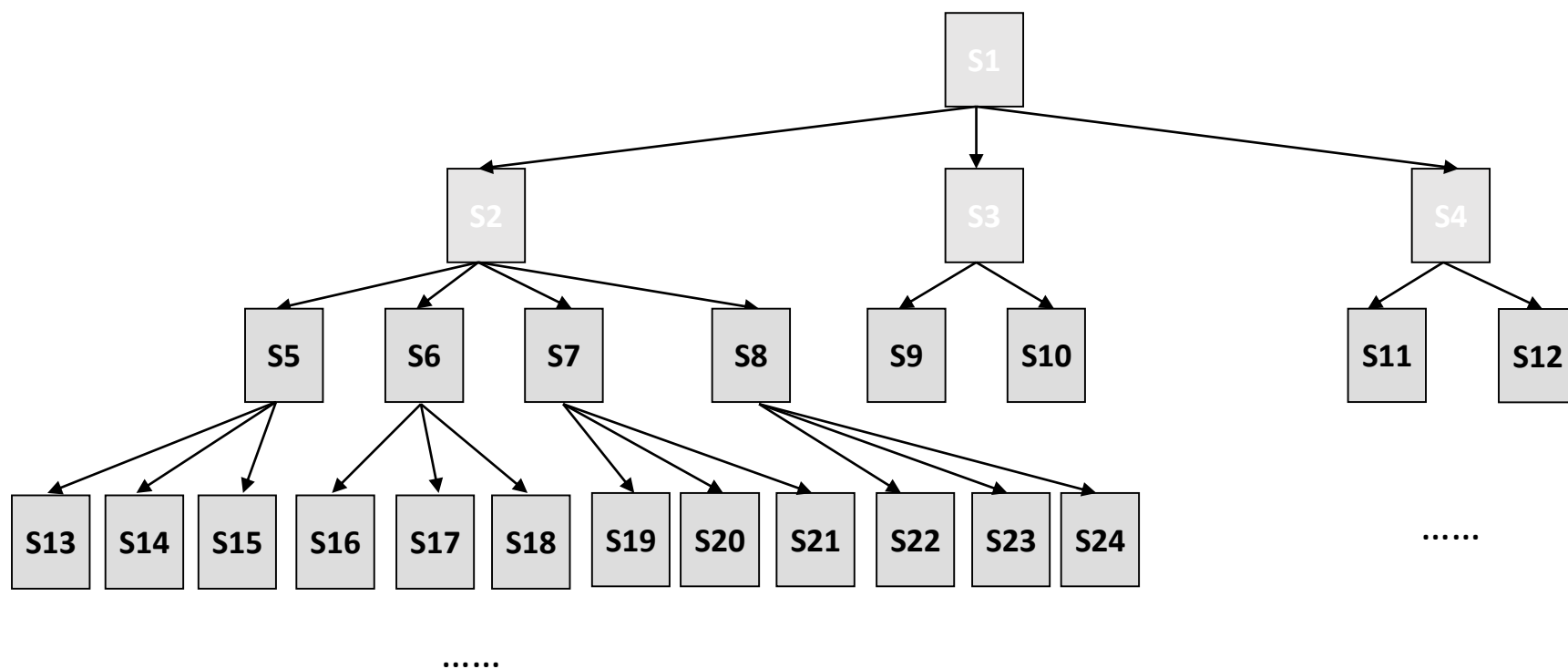
基本思想：

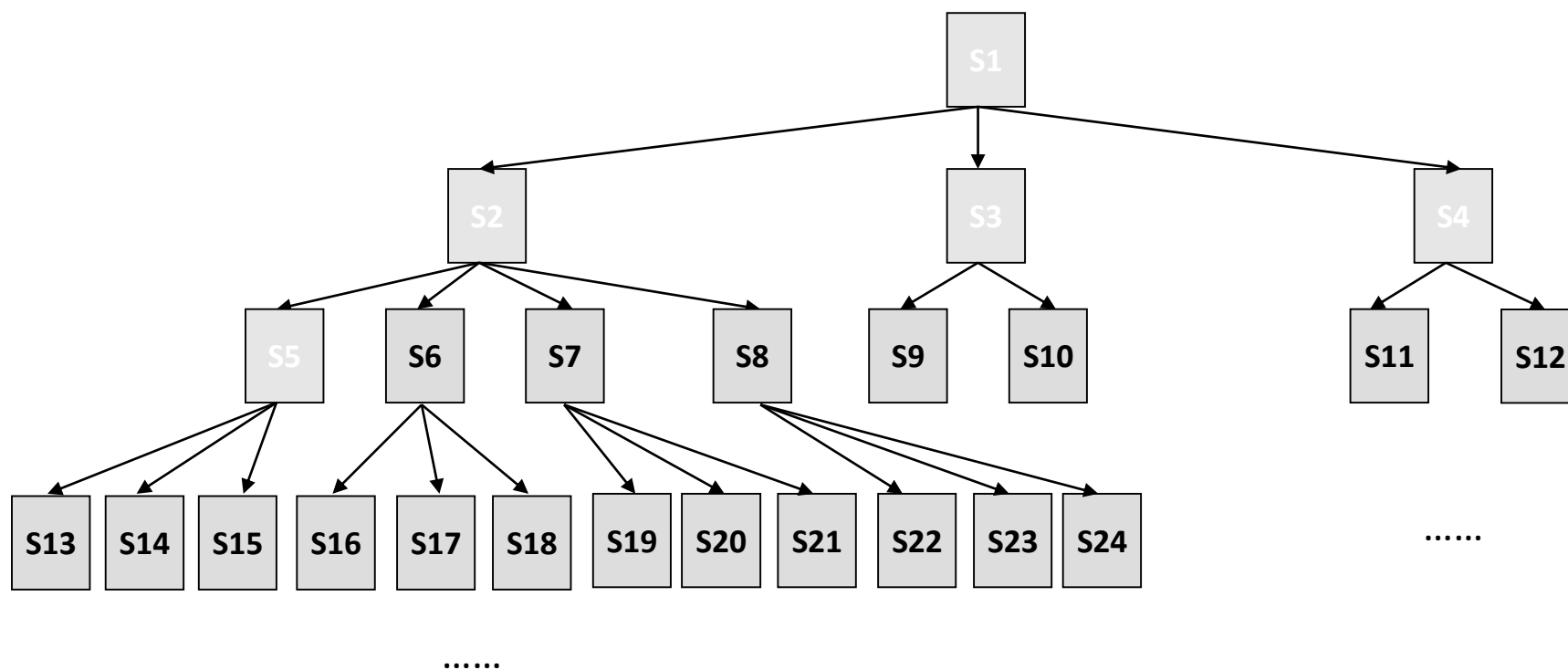
- 1.从初始状态 $S$  开始，生成初始状态可以到达的所有状态。
- 2.依次枚举这些状态，检查是否出现目标状态 $G$ 。
- 3.若未出现，就对该层所有状态节点，顺序扩展本节点可以到达的所有状态。
- 4.生成再下一层的所有状态节点，对这一层的所有状态节点检查是否出现 $G$ ，若未出现，继续按上面思想生成再下一层的所有状态节点，这样一层一层往下展开。
- 5.直到出现目标状态为止。

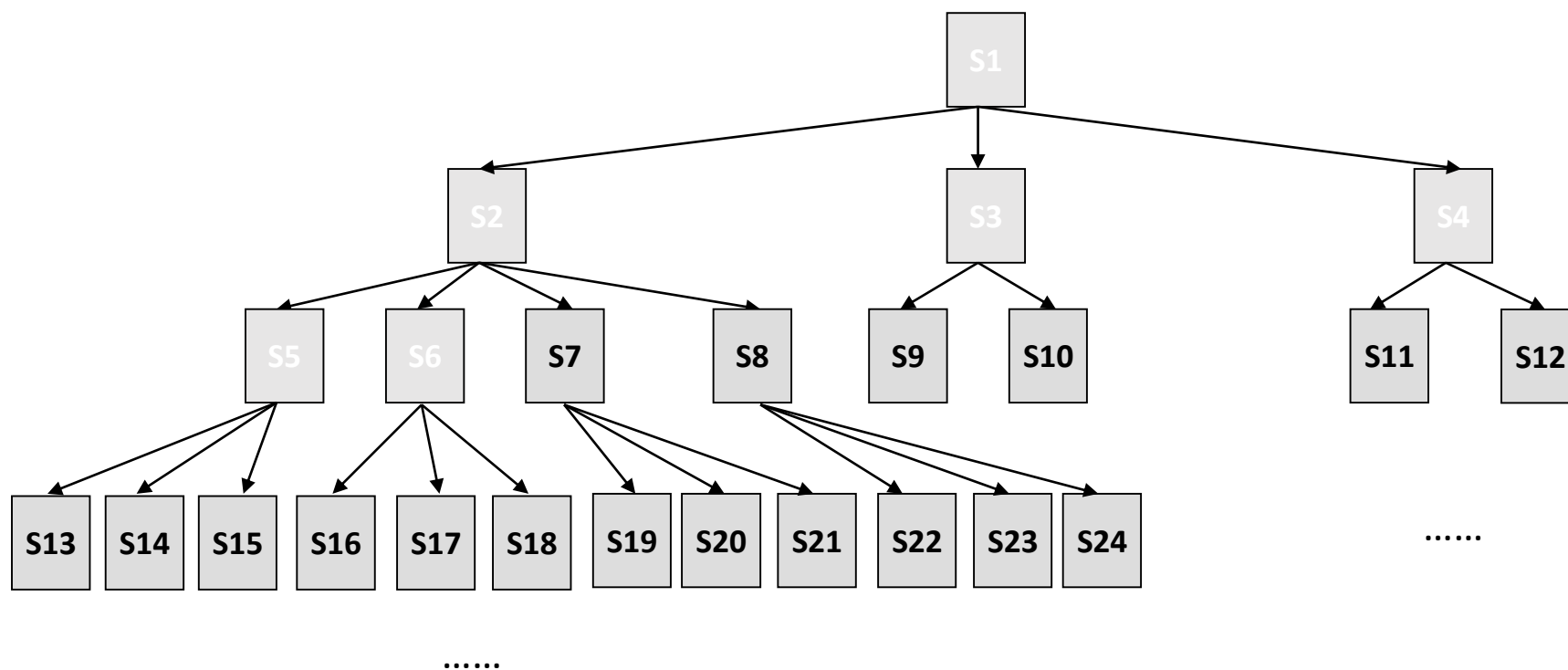




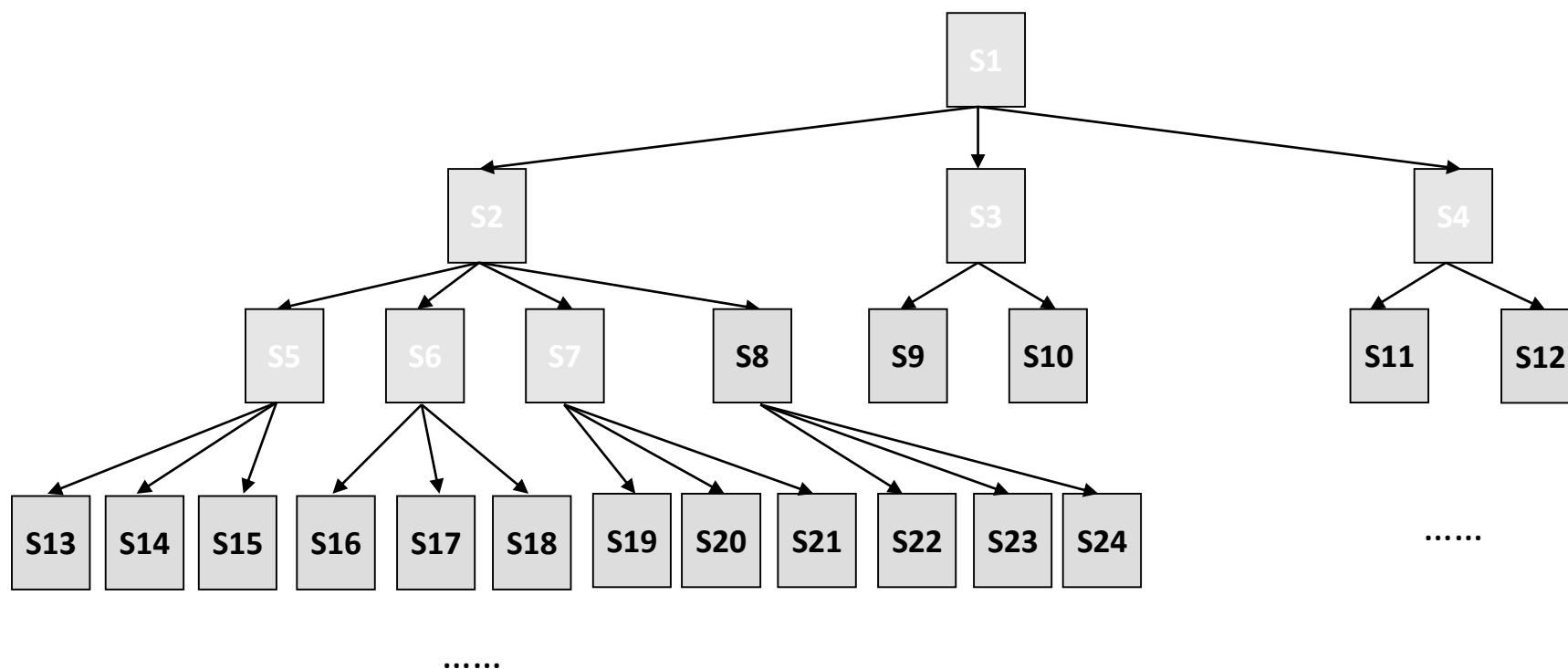


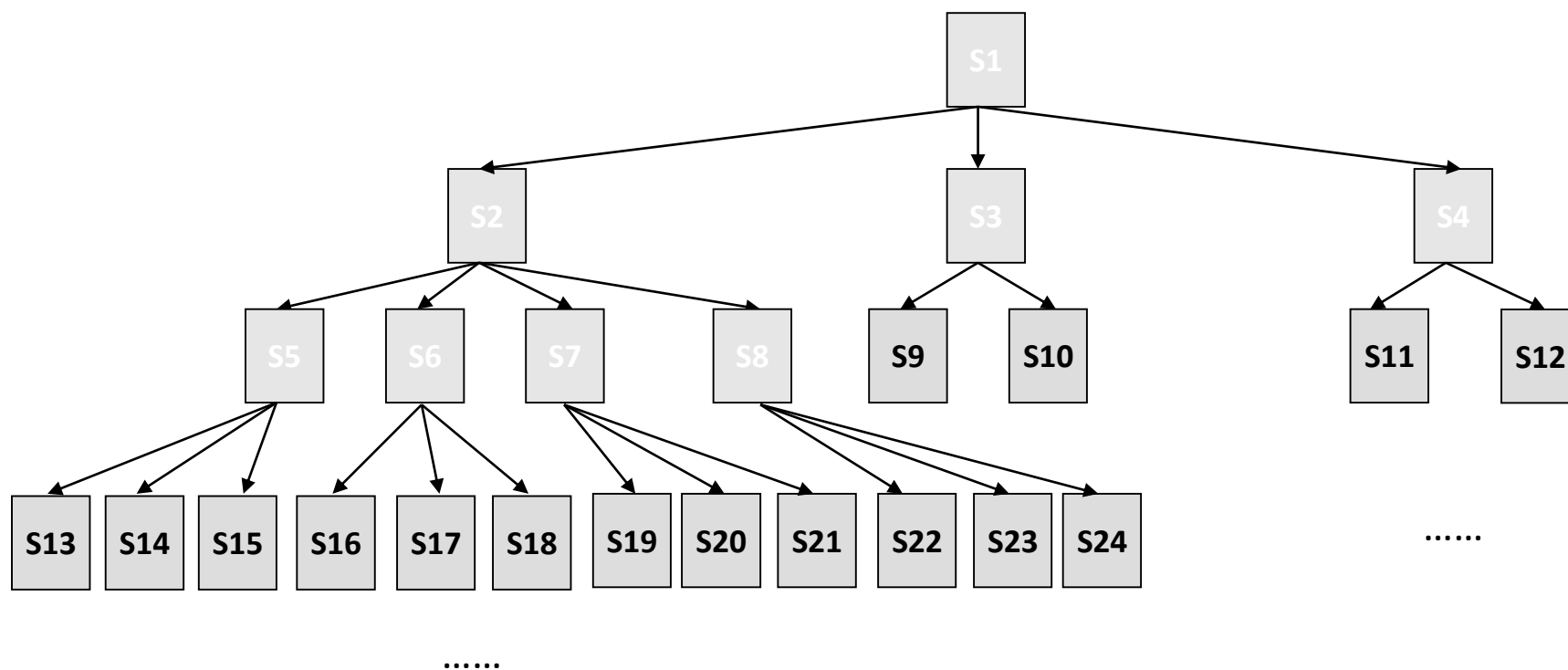


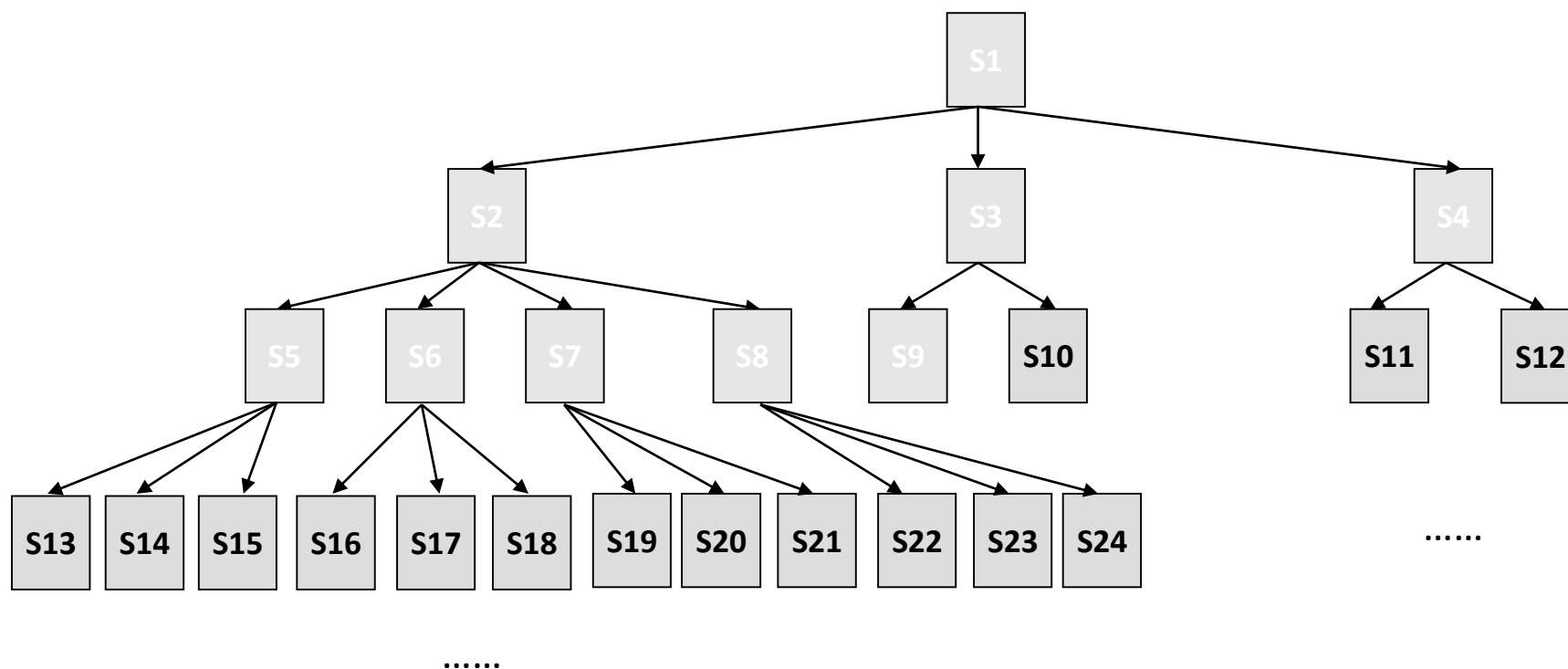


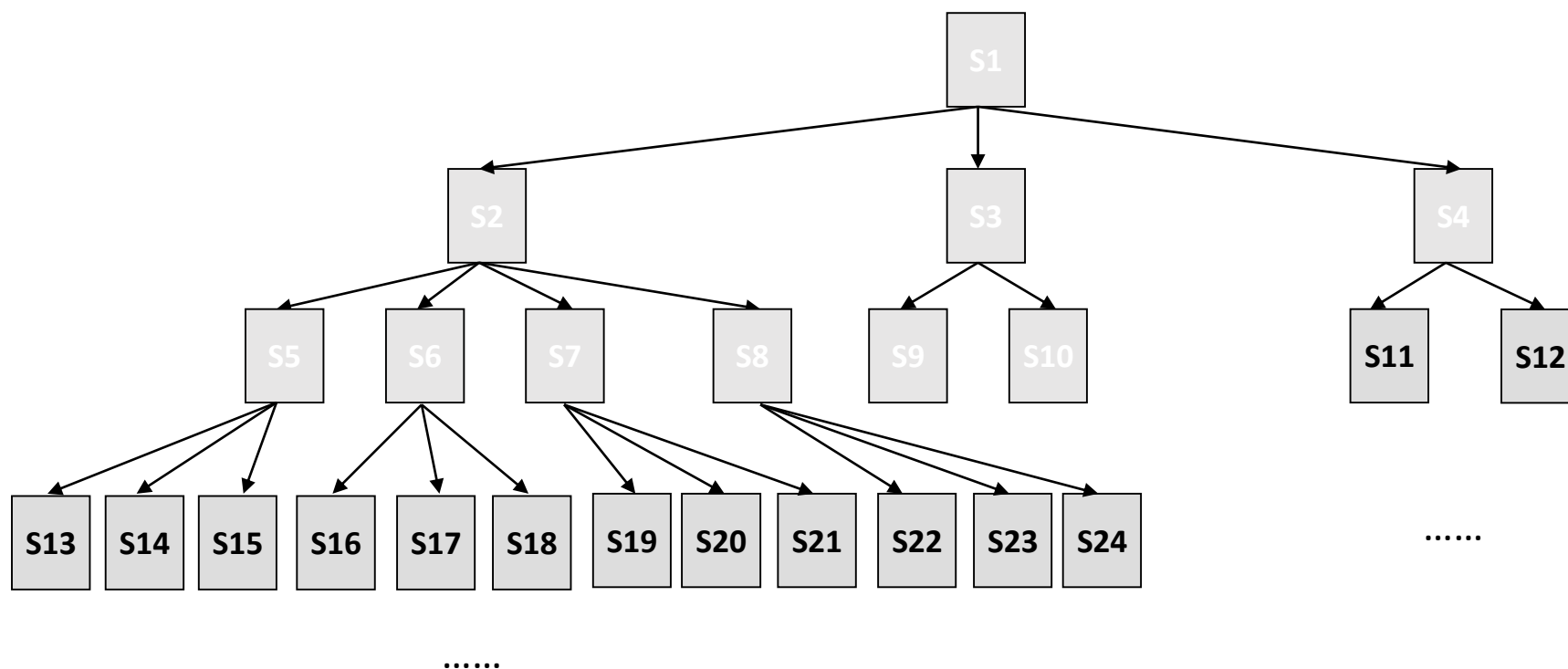


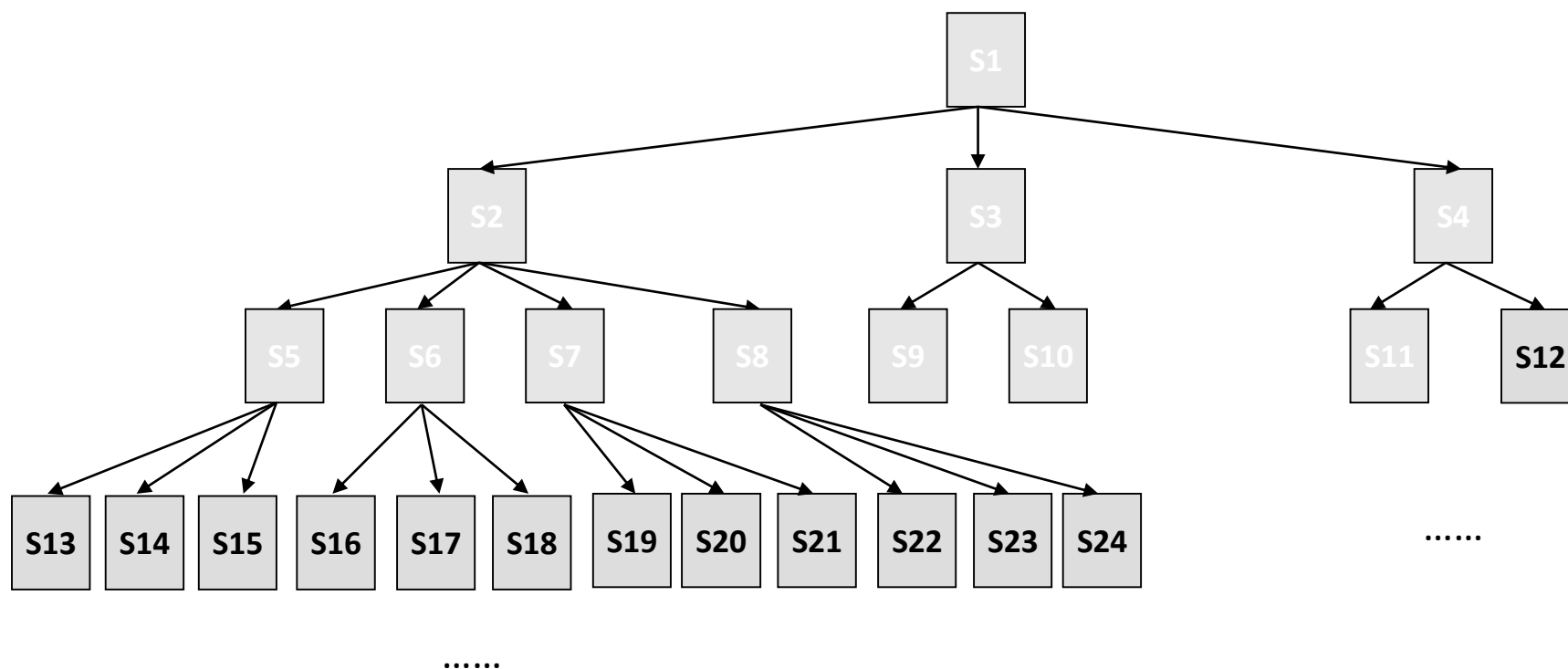


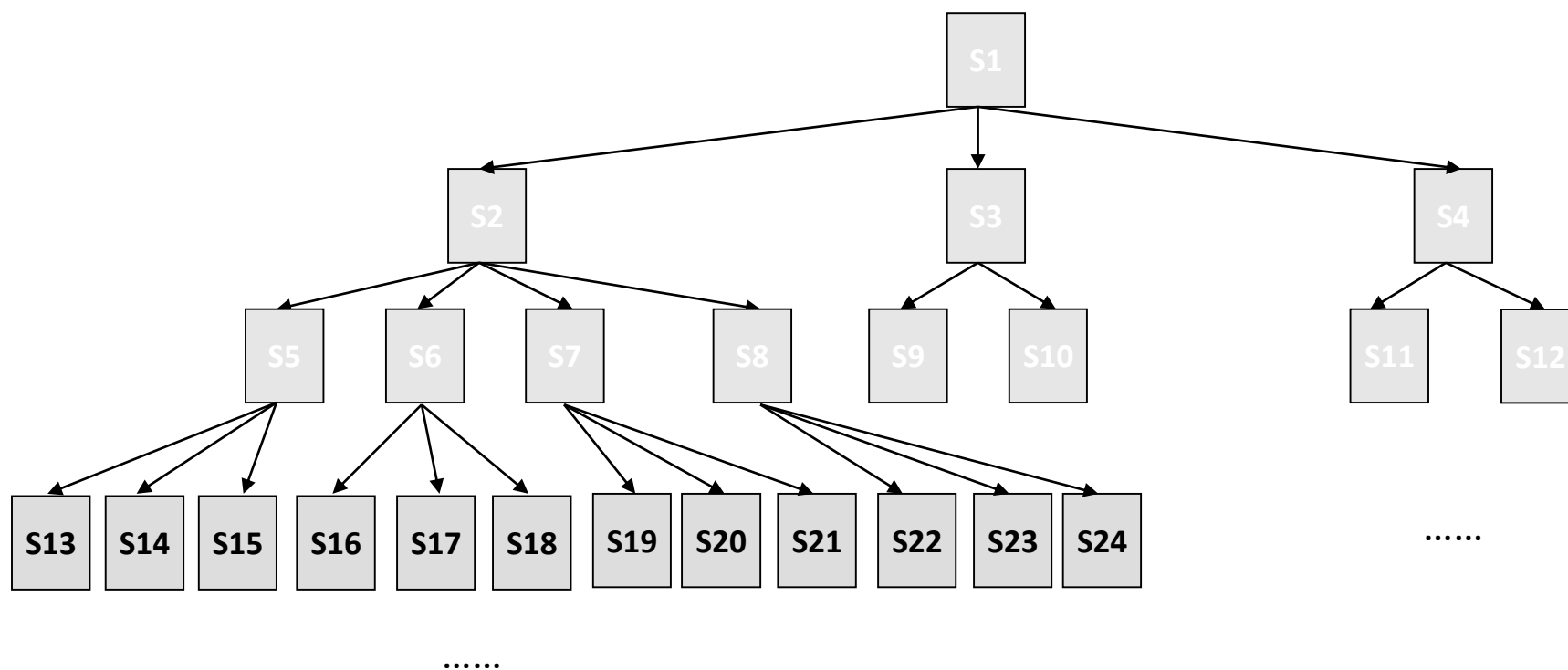


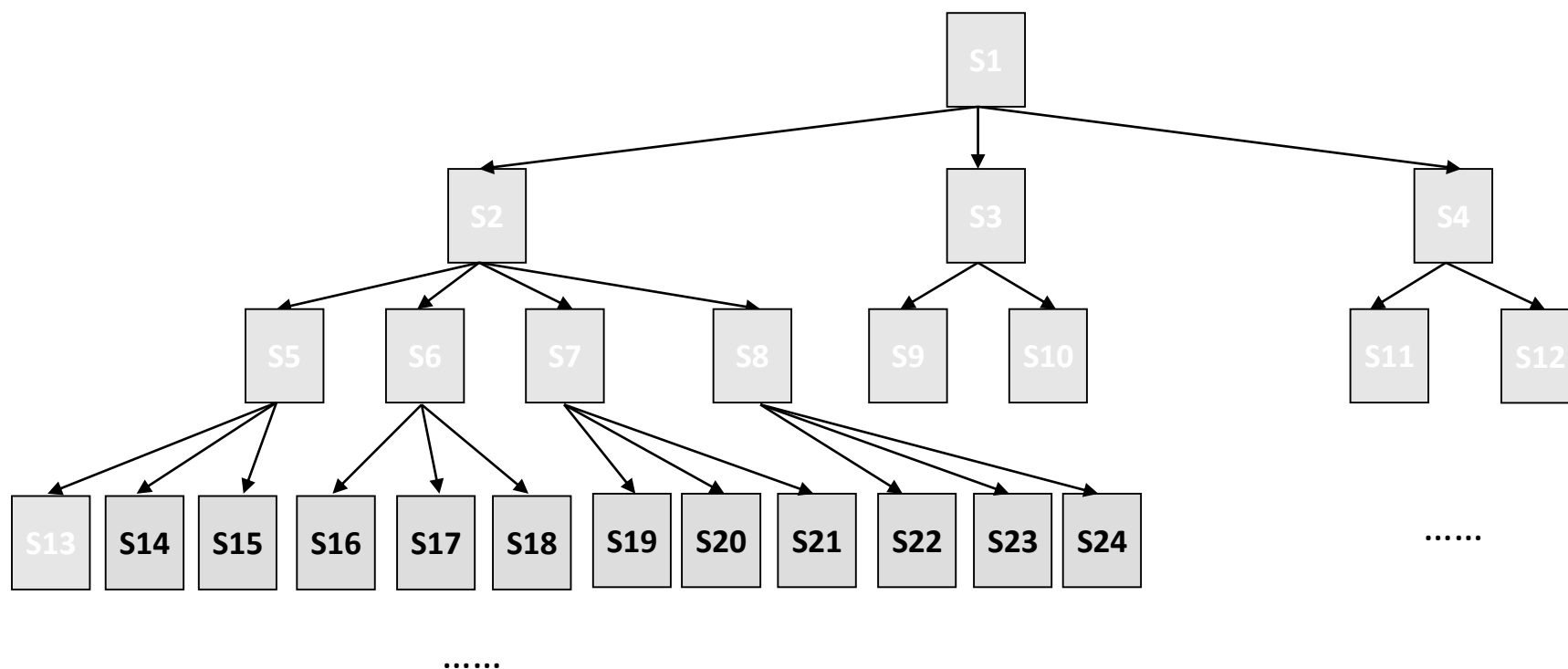


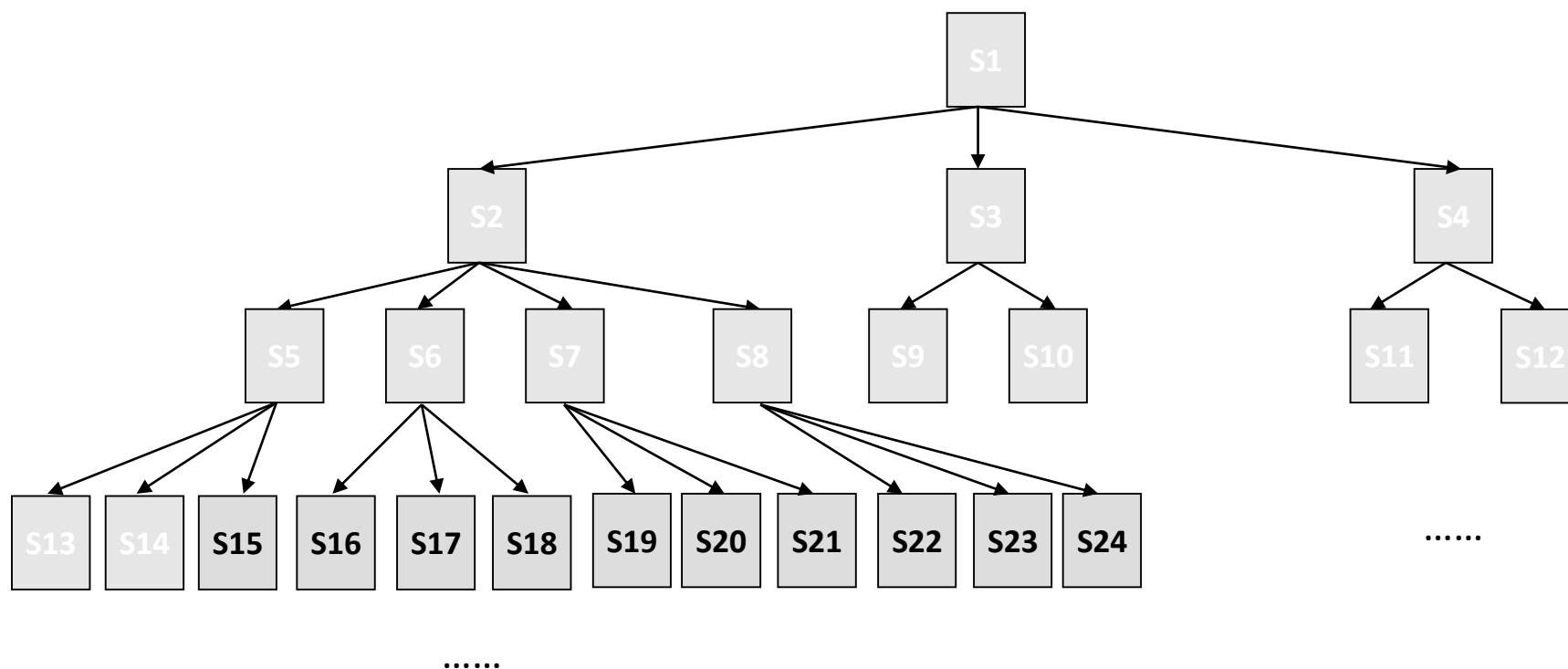




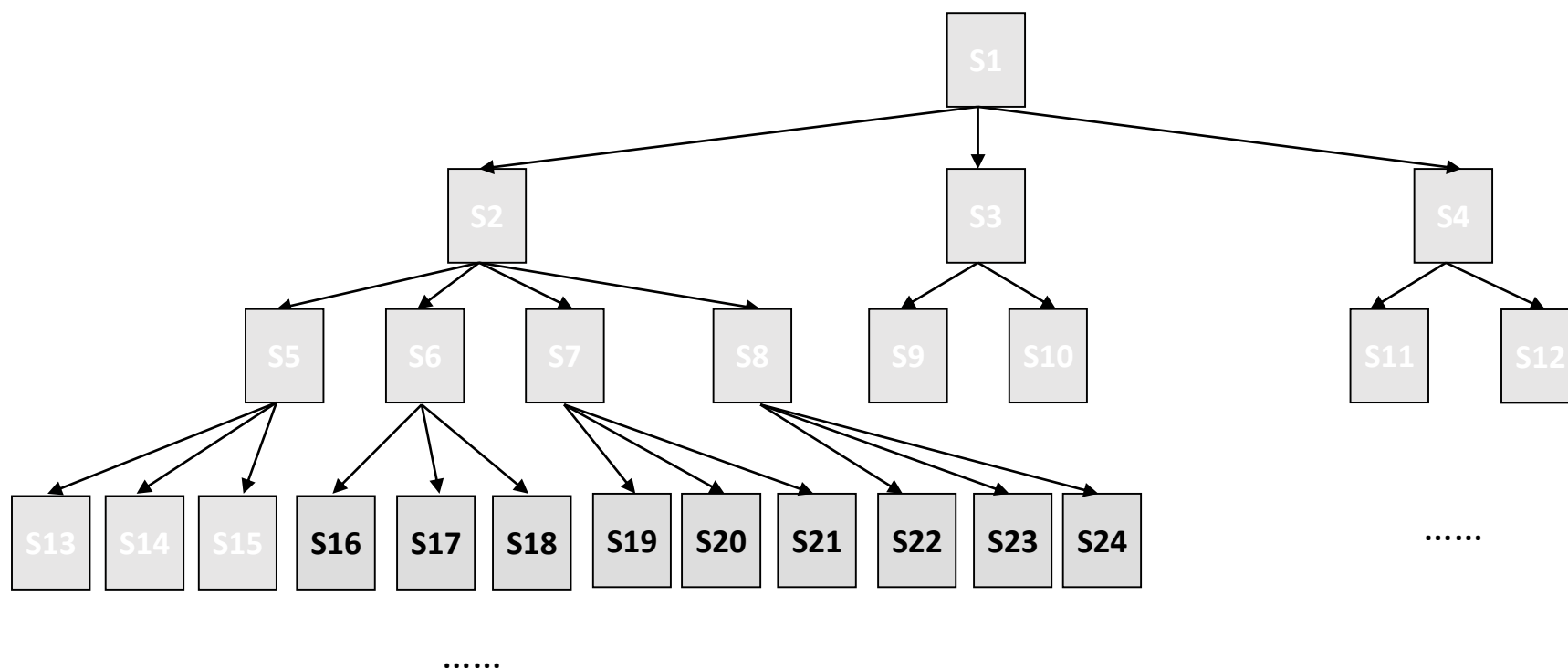


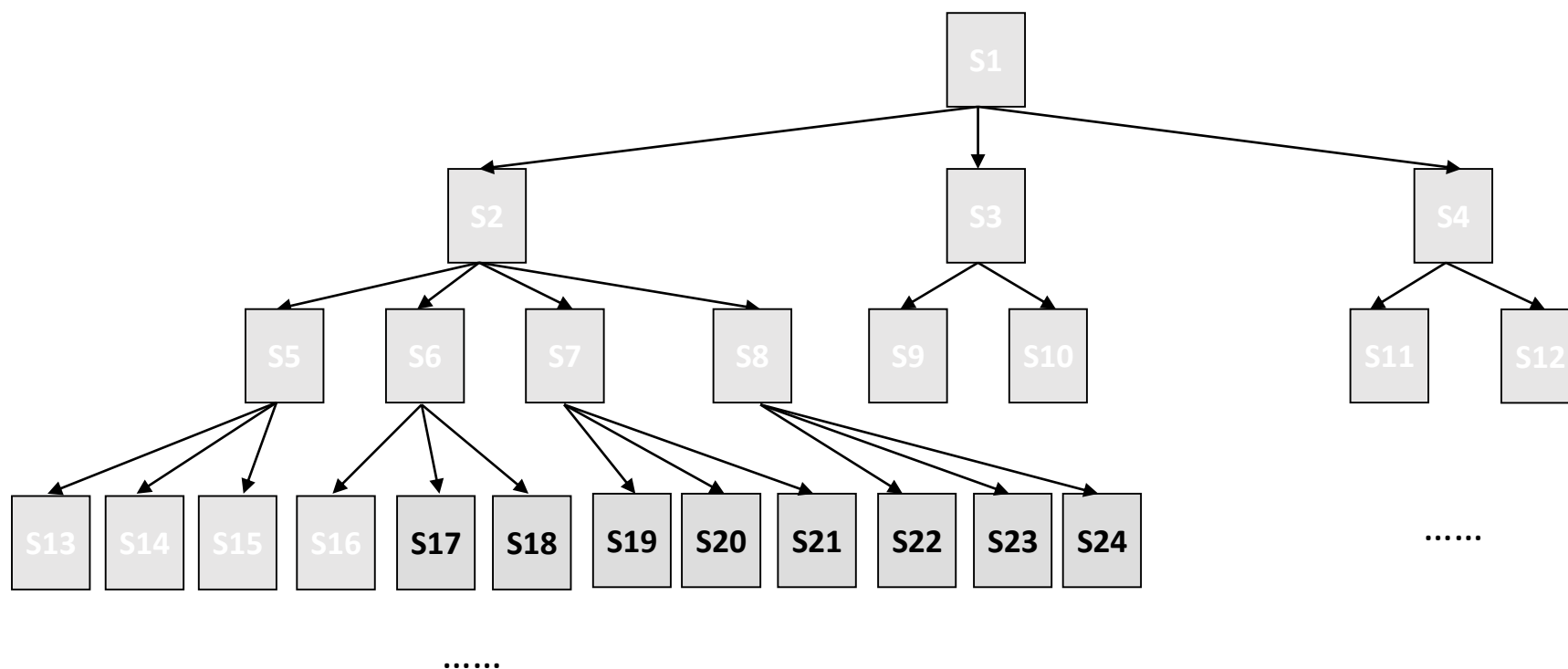


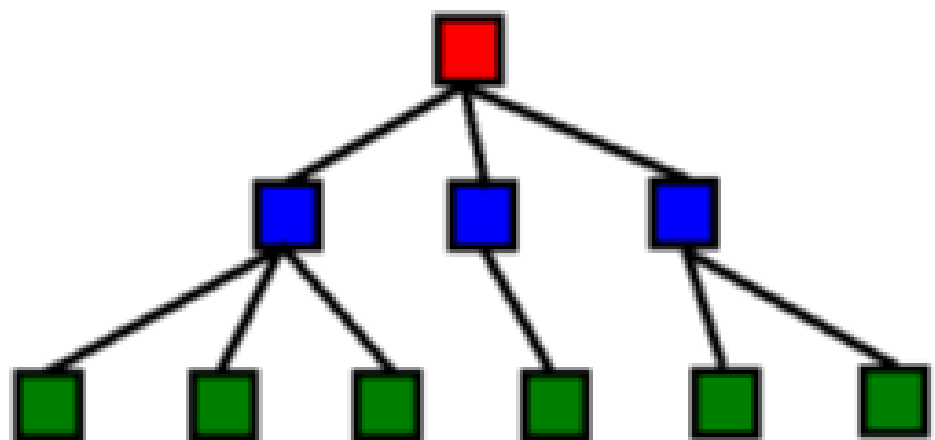












左图每个方块表示一个状态，浅蓝色的表示遍历了该状态。

广度优先搜索即是按层数一层一层来遍历，先将一层全部扩展，然后再进行下一层。

利用队列先进先出（**FIFO**）的性质恰好可以来完成这个任务

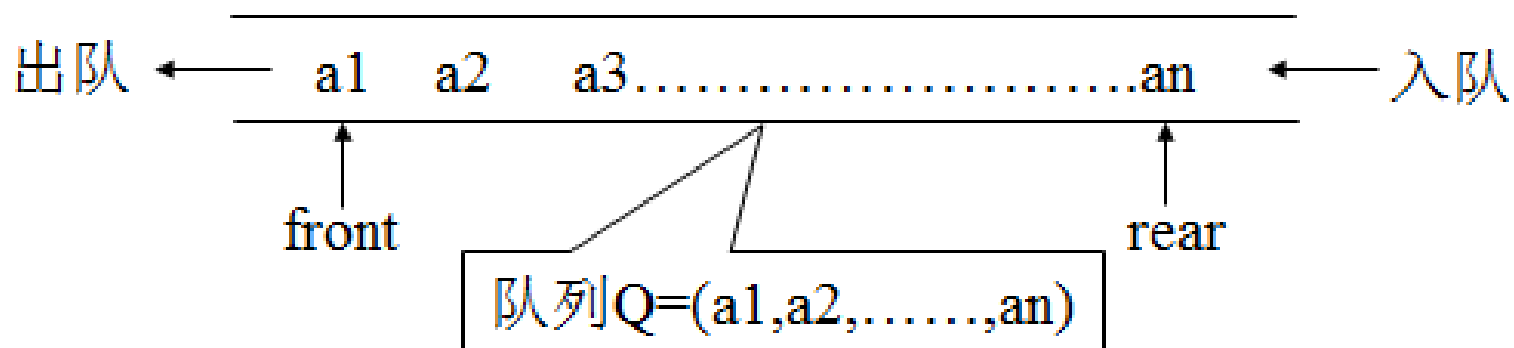
对应的队列的情况





具体过程：

- 1 每次取出队列首元素（初始状态），进行拓展
- 2 然后把拓展所得到的可行状态都放到队列里面
- 3 将初始状态删除
- 4 循环执行以上三步直到找到目标状态或者队列为空。





**具体的迷宫问题：**迷宫大小 $n*m$ ，用字符串表示迷宫。其中，#表示墙，\*表示路径，S表示起点，T表示终点。

**Input**

5 6

###T#\*

S#\*\*\*\*\*

\*##\*##

\*\*\*\*\*

\*#\*#\*#

**Output**

(1,0)->(2,0)->(3,0)->(3,1)->(3,2)-  
>(3,3)->(2,3)->(1,3)->(0,3)

###T#\*

S#\***\***\*\*\*

**\***##**\***##

**\*****\*****\*****\*****\*****\***

\*#\*#\*#



# 今天你学会了什么？

- 一、深度优先搜索
- 二、广度优先搜索
- 三、实践出真知



掌握搜索算法基本思想。

分别使用深度优先搜索和广度优先搜索完成迷宫问题。



- 一分耕耘，一分收获！
- 希望在今后的学习生活中共同进步！

# 谢谢！