

# 算法设计与分析

---

## ➤ LECTURE 10

# Outline



## Lecture 10

### 回溯与分支限界法

- 回溯法
  - $n$  皇后问题
  - 装载问题
  - 0-1 背包问题
  - 旅行商问题
- 分支限界法
  - 装载问题
  - 背包问题

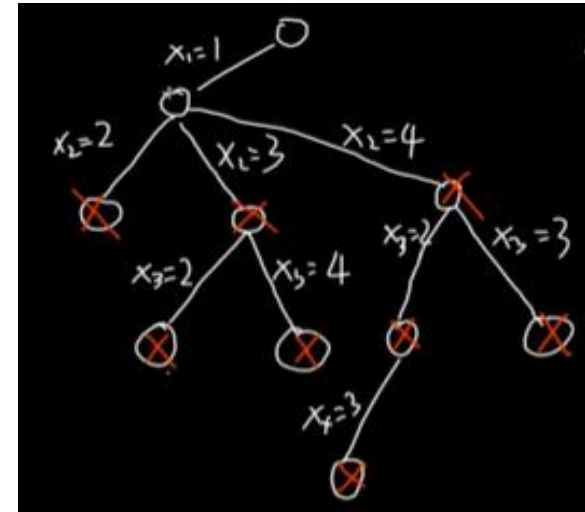
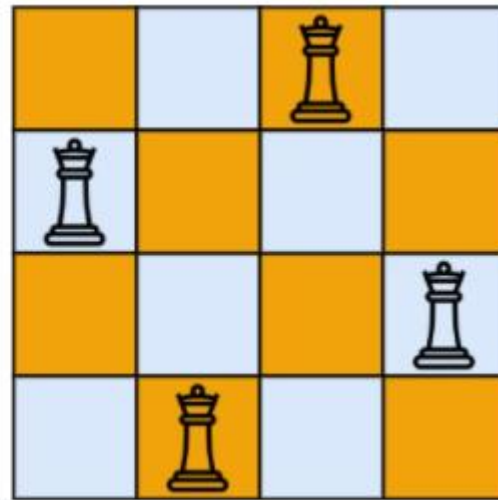
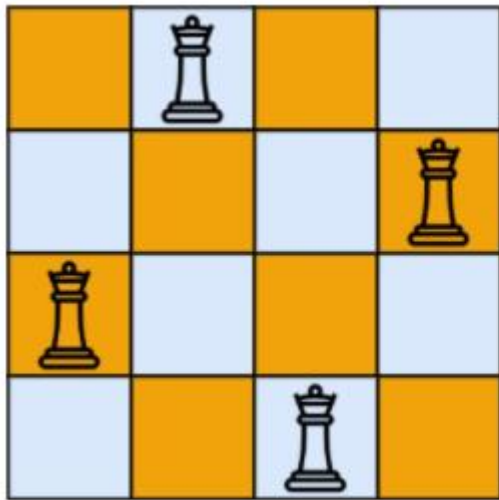
# 回溯法



- 把解空间规划成一棵解空间树，在求解过程中使用剪枝策略来提高搜索效率
- 常用于求解搜索问题和优化问题
- 穷举所有的“状态空间”（树）
  - 路径（根到叶子）对应一个可行解
  - 求解问题过程就是在状态树上遍历的过程
  - DFS/BFS
- 回溯法：在约束条件下对解空间树进行深度优先查找的过程，并在查找过程中减去那些不满足条件的分支。

# N 皇后问题

- 输入:  $n \times n$  的格子,  $n$  个皇后
- 输出: 在所有的皇后不能互相攻击 (同行, 同列, 同对角线) 的情况下, 把  $n$  个皇后全部摆放在  $n \times n$  的格子中




<https://leetcode-cn.com/problems/n-queens/>

# N 皇后问题



## □ 伪码

A small hand-drawn tree diagram with a root node and three child nodes, each containing a circle with an 'X' inside.

```
bool solve_queen(row)
    if row >= size then done
    for col = 0 to n-1
        if (row, col) is a safe position
            place a queen at (row, col)
            if solve_queen(row+1) is True
                return True
            else
                remove queen from (row, col)
    return False
```

回溯

[https://www.bilibili.com/video/BV1fQ4y1A7Aq?spm\\_id\\_from=333.999.0.0](https://www.bilibili.com/video/BV1fQ4y1A7Aq?spm_id_from=333.999.0.0)



# 分支限界法

- 常用于求解优化问题
- 穷举所有的“状态”，利用上界/下界来提高搜索效率
- 术语
  - 活结点：该节点刚被生成，子节点还未产生
  - 当前节点 E-node：当前正在展开的节点
  - 死节点：不需要展开的节点
- 搜索方法
  - E-node 需要确定优先被展开的节点（优先级）

# 分支限界法



- **上界函数**：以当前节点为根的可行性解可能达到的极值
- **限界值**：搜索到某一节点时，已经得到的可行解的最优值
  
- **上界**：对当前分支的一种预测，是选择当前分支后可能达到的最大值，不一定是可行性解的值。
- **限界**：已找到的可行性解当中最优解的值。
  
- 若某分支的上界小于限界，说明在分支当中不可能找到小于该限界的可行性解，则应当停止对此分支的继续搜索。
  - 增加了上界函数与界限值的约束，通常可以剪掉更多的分支，也因此被称为分支限界法。



# 分支限界法的基本步骤

---

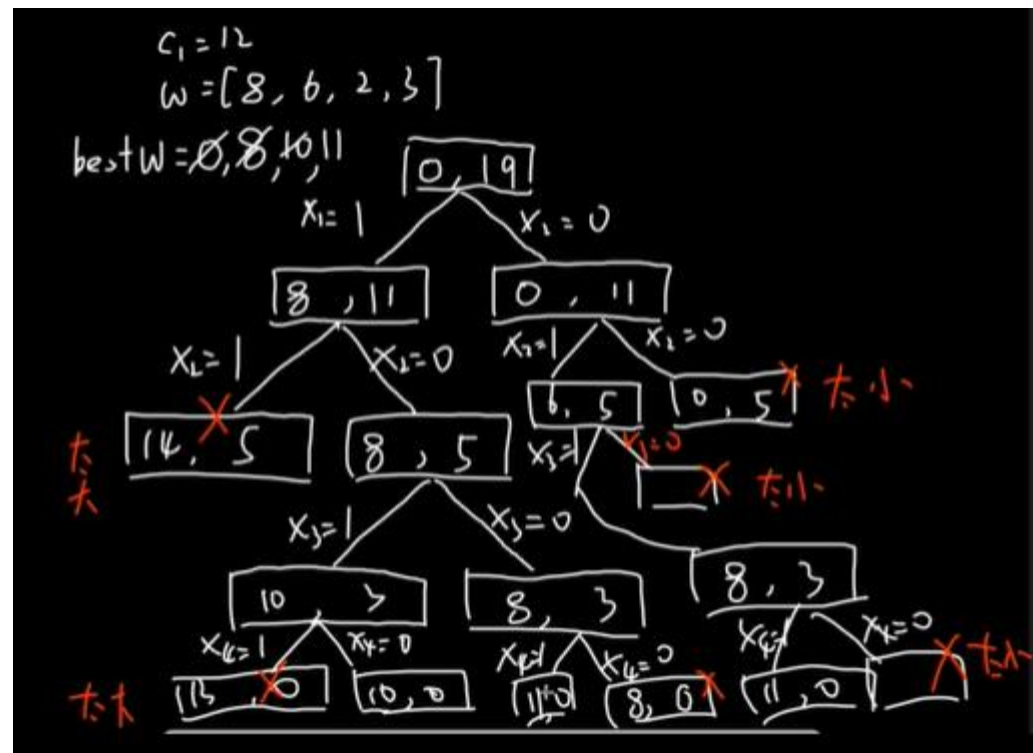
- 建立上界函数
- 求得限界值
- 根据剪枝条件，停止分支的搜索，向上回溯到父节点
  - 节点上界小于界限值
  - 违法约束条件
  - 叶子节点
- 限界值的更新





# 装载问题

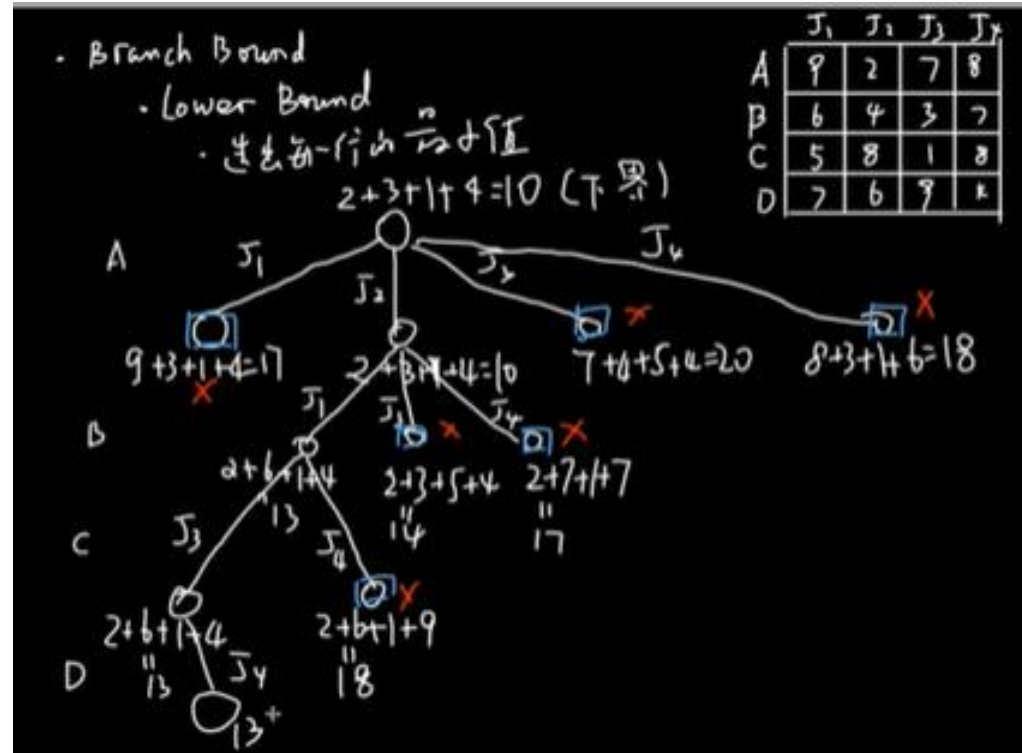
- 输入:  $n$  个物品重量, 总容量
- 输出: 最大装载重量
- 状态节点保持数据
  - $cw$ : 当前装载数
  - $r$ : 剩余物品的累加重量
- $bestw$ : 当前最优装载数
- 剪枝函数
  - $cw + r \leq bestw$
  - $cw > c$
- 优先级





# 任务分配

- 输入:  $N$  workers,  $N$  jobs,  $n \times n$  time cost.
- 输出: 总耗时最小的分配方案



[https://www.bilibili.com/video/BV1CQ4y1P7qc?spm\\_id\\_from=333.999.0.0](https://www.bilibili.com/video/BV1CQ4y1P7qc?spm_id_from=333.999.0.0)

Thanks