$$O(1)<O(\log n)<O(n)<O(n\log n)<O(n^2)<O(n^3)$$
$$O(2^n)<O(n!)<O(n^n)$$

渐近记号	名称
$T(n)=\Theta(g(n))$	渐近紧确界
$T(n)=O(g(n))$	渐近上界
$T(n)=\Omega(g(n))$	渐近下界

课件各个复杂度

1. 动态规划
    1. 01背包
        1. 暴力枚举：2^n
        2. 递推计算： C.n
    2. 最大子数组
    3. 最长公共子序列
    4. 最长公共子串
    5. 编辑距离问题
    6. 钢条切割 东一大公公巨刚
2. 分治法
    1. 归并排序 nlogn
    2. 递归式
    3. 最大子数组
    4. 逆序对计数
    5. 快速排序
    6. 次序选择 粉笔弟打会计者
3. 贪心法

1. 部分背包
2. 活动选择
3. 霍夫曼编码 贪杯动漫

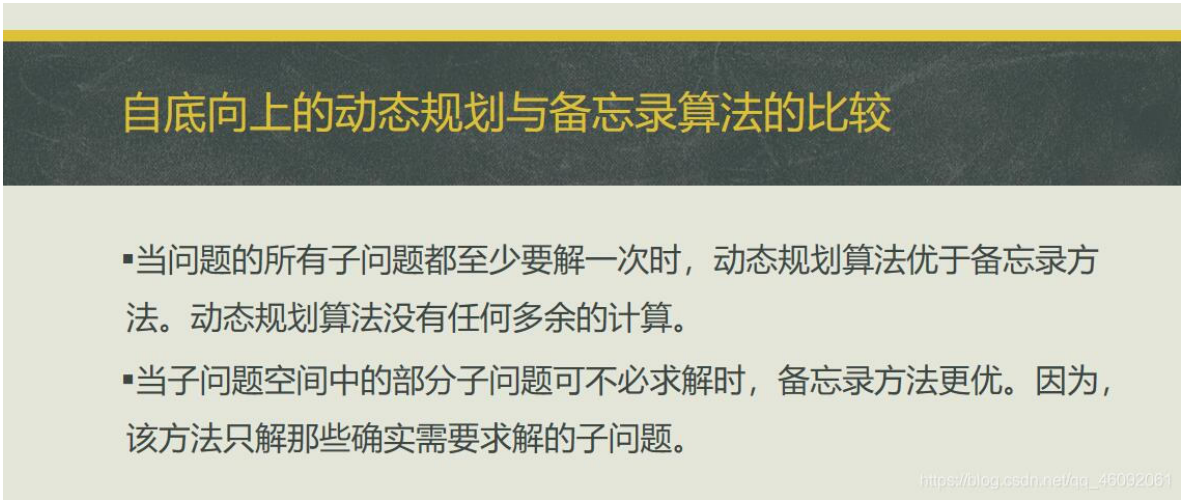
1.动态规划

动态规划是先用**暴力枚举**(自顶向下)将所有的可能性列举出来，然后为了避免大量重复的计算浪费时间，使用哈希/字典等等缓存数据，这叫做**记忆化搜索或者剪枝**；接着改写成类似数学归纳法的**迭代形式**(自底向上)

2.贪心

贪心是在**当前条件下的最优解**推导出**全局最优解**，这类题型没有固定模板，如果能用局部推导全局那就可以

1. 分治法的设计思想是将一个难以直接解决的大问题分割成规模较小的子问题，分别解决子问题，最后将子问题的解组合起来形成原问题的解。这要求原问题和子问题 **问题规模不同，问题性质相同**
2. 01背包用动态规划，背包问题用贪心
3. 具有最优子结构的算法有 **动态规划法、贪心算法、分治算法**
4. 算法分析中，记号 **O** 表示 **渐进上界**
5. 备忘录算法是**动态规划算法**的变形
6. 用贪心法求解背包问题时的贪心选择策略是 **单位容量带来的价值之比**
7. 算法的性质包括**输入、输出、确定性、有穷性、可行性**
8. 最长公共子序列算法利用的算法是 **动态规划算法**
9. 若总是以待排序列的第一个元素作为基准元素进行快速排序，那么最坏情况下的时间复杂度为 **O(n^2)**
10. 贪心算法与动态规划算法的主要区别是 **贪心性质的选择**：最优子结构、构造最优解、定义最优解
11. 分治算法一定 由三个步骤组成：**划分子问题、求解子问题、合并子问题的解**
12. 若一个问题既可以用迭代方式也可以用递归方式求解，则**迭代**方法具有更高的时空效率
13. 迪杰斯特拉 (Dijkstra) 算法按照路径长度递增的方式求解单源点最短路径问题，该算法运用了**贪心**算法策略
14. 动态规划的基本要素为 **最优子结构与重叠子问题性质**
15. 贪心求解最优解问题，一般重要性质：**最优子结构性质与贪心性质的选择**，贪心算法不一定满足最优子结构的性质
16. **动态规划法** 通常以自底向上的方式求解最优解
17. 衡量一个算法好坏的标准是 **时间复杂度低**
18. 实现循环赛日程表利用的算法是 **分治策略**
19. P类问题包含在NP类问题中
20. **最优子结构性质**是贪心算法与动态规划算法的共同点
21. 用分治法的三个步骤是 **划分 解决 合并**

22. 动态规划算法有一个变形方法 **备忘录。**这种方法不同于动态规划算法“自底 向上”的填充方向，而是“自顶向下”的递归方向，将每个解过的子问题存储下来以 备需要时查看，同样也可避免相同子问题的重复求解
23. **背包问题与0-1背包问题**的不同点在于，在选择物品装入背包时，可以只选择物品的一部分， 而不一定要选择物品的全部
24. 动态规划中存储子问题是为了 **避免重复求解子问题**
25. 从分治法的一般设计模式可以看出，用它设计出的程序一般是 **递归算法**
26. 简述分治法和动态规划算法的相同点和不同点
    相同点：二者要求原问题具有最优子结构，都是将问题分而治之分解成若干个规模较 小的子问题；
    不同点：分治法将原问题分解成独立的子问题，然后递归求解子问题，并组合成原问题的解。而动态规划应用于子问题不独立时，它的实质是分治思想和解决冗余，为避免重复计算，它将已经计算过的子问题存储起来，达到最优解决问题的目的。
27. 贪心算法与分治法和动态规划算法的异同
    1）分治法将原问题分解成独立的子问题，然后递归求解子问题，并组合成原问题的解。
    2）动态规划应用于子问题不独立时，它的实质是分治思想和解决冗余，为避免重复计算，它将已经计算过的子问题存储起来，达到最优解决问题的目的。
    3）贪心法与动态规划法和分治法类似，都是将问题分解为规模更小的、相似的子问题，并通过求解子问题产生一个最优解。
    有些具有最优子结构性质的问题，可以用动态规划算法求解，但是用贪心算法更简单、更直接，且解题效率更高
28. 动态规划算法通常以自底向上的方式解各子问题，而贪心算法则通常以自顶向下的方式进行
29. 

30. 贪心算法、动态规划算法、回溯算法、分支限界算法都可以得到最优解中最高效的是贪心算法
31. P类是容易处理的，NP问题是能被处理但是有点难的，p类是否==NP目前没有定论
32. 动态规划是**自底向上**方法的，动态规划中的备忘录是**自顶向下**
33. 实现循环赛日程表、棋盘覆盖、Strassen矩阵乘法利用的算法是分治法
34. 哈弗曼编码的贪心算法所需的计算时间是 O(nlogn)
35. **P类问题包含在NP类问题中**