

四、  
八

得分  
阅

算法 A 和算法 B 解同一问题，设算法 A 的时间复杂性满足递归方程

$$\begin{cases} T(n) = 1, & n = 1 \\ T(n) = 4T(n/2) + n, & n > 1 \end{cases}$$
， 算法 B 的时间复杂性满足递归方程

$$\begin{cases} T(n) = 1, & n = 1 \\ T(n) = aT(n/4) + n, & n > 1 \end{cases}$$
， 若要使得算法 A 时间复杂性的阶高于算法 B 时间复杂性的阶，a 的最大整数值可取多少？

给定一个  $2 \times N$  的棋盘格，现在用  $1 \times 2$  的特殊方格去覆盖它，有几种方式？

要求如下：

描述求解该问题的思路；

(5 分) 列出递推公式。

得分	
阅卷人	

四、(本题满分 25 分)

战争时期经常需要用电报传递信息。为了更快地传递信息，可以采用对信息进行压缩的方法减少信息长度。假设我们需要用二进制编码发送以下字符：“Traitor appears, action cancelled”，问如何对这串字符进行编码才能使得其编码长度最短？

要求如下：

- (1) (10 分) 简述解决该问题的贪心选择性质；
- (2) (15 分) 证明你的贪心选择性质的正确性。

得分	
阅卷人	

五、(本题满分 25 分)

某港口到了一批货物，共 150 吨，要求将这些货物分别装到 2 辆载重量分别为 70 吨和 80 的货车上（假设这些货物正好能够装到这两辆货车上）。

要求如下：

- (1) (10 分) 描述分支限界法和回溯法的异同。
- (2) (10 分) 给出求解上述问题的约束函数和上界函数。
- (3) (5 分) 如何操作可以使右子树测试尽快生效？

得分	
阅卷人	

公司老板有一袋金块，用于奖励优秀员工。每个月有两名表现最好的员工分别被奖励 1 个金块。奖励规则是：排名第一的员工将得到袋中最重的金块，排名第二的员工将得到袋中最轻的金块。根据这种方式，除非有新的金块加入袋中，否则第一名员工所得到的金块总是比第二名员工所得到的金块重。如果有新的金块周期性的加入袋中，则每个月都必须找出最轻和最重的金块。假设有一个天平可以用来称量金块重量，希望用最少的比较次数找出最轻和最重的金块。要求如下：

- (1) (10 分) 给出两种求解该问题的算法思路或伪代码；
- (2) (10 分) 分析你给出的算法时间复杂度（要求有分析过程）。

二 (1) 分治法：将金块分成两个等分的部分A和B。对A和B分别进行称重，得到A的总重量和B的总重量。如果A的总重量大于B的总重量，说明最轻的金块在B中，最重的金块在A中。将B分成两个等分的部分，继续执行步骤2。如果B的总重量大于A的总重量，说明最轻的金块在A中，最重的金块在B中。将A分成两个等分的部分，继续执行步骤2。当只剩下两个金块时，比较这两个金块的重量，较轻的为最轻金块，较重的为最重金块。分组策略：将金块分成 $n/2$ 组，每组2个金块。

对每组中的两个金块进行称重比较。找出每组中较轻的金块和较重的金块。对所有较轻的金块进行一次称重，得到最轻金块。对所有较重的金块进行一次称重，得到最重金块。

(2) 时间复杂度： $O(n \log n)$ 、 $O(n)$

四、(1) 构建字符频率表：统计每个字符在待发送的字符串中出现的频率。构建霍夫曼树：根据字符频率表构建霍夫曼树。霍夫曼树是一种最优前缀编码树，频率较高的字符将具有较短的编码，频率较低的字符将具有较长的编码。生成编码表：从霍夫曼树的根节点开始，沿着左子树走为0，沿着右子树走为1，将路径上的编码记录下来。为每个字符生成对应的编码表。进行编码：根据生成的编码表，将待发送的字符依次进行编码，替换为对应的二进制编码。这样，通过霍夫曼编码，字符出现频率较高的部分将被压缩为较短的二进制编码，字符出现频率较低的部分将被压缩为较长的二进制编码，从而使得整体编码长度最短。

四 (2) 贪心选择性质的简述：霍夫曼编码是一种基于贪心选择的算法。其贪心选择性质表现在构建霍夫曼树的过程中，每次都选择频率最低的两个字符（或子树）进行合并。通过不断选择频率最低的字符，保证了生成的编码树具有最小的平均编码长度，从而达到最短编码的目的。

证明贪心选择性质的正确性：为了证明霍夫曼编码的贪心选择性质的正确性，需要证明每次合并频率最低的字符或子树，所得到的编码树仍然是最优的。

### 三.

(1) 求解该问题的思路：对于给定的 $2^*N$ 的棋盘格，我们可以考虑使用动态规划的方法来求解覆盖方式的数量。动态规划是一种利用已知结果推导出未知结果的算法，通过将问题分解为子问题并存储子问题的解来避免重复计算。可以从左到右依次考虑每一列，对于每一列，有两种放置方式：横着放置两个 $1^*2$ 的方格，这样当前列和下一列都会被覆盖；竖着放置一个 $1^*2$ 的方格，这样当前列会被覆盖，下一列不会被覆盖。

基于这两种放置方式，我们可以得出以下递推关系：

(2) 递推公式：假设用 $dp[i]$ 表示 $2^*i$ 的棋盘格的覆盖方式数量，则递推公式为： $dp[i] = dp[i-1] + dp[i-2]$

一.根据主定理，算法A的时间复杂度为 $O(n^2)$ ，所以 $a$ 最大为7，B时间复杂度为 $O(n^{(7/4)})$

### 五.

(1) 分支限界法和回溯法的异同：

相同点：

都是求解组合优化问题的算法，通过搜索解空间来找到满足问题约束的最优解。

都涉及到对解空间的遍历和搜索。

都可以用于求解组合优化问题中的可行解或最优解。

不同点：

回溯法是一种深度优先搜索算法，通过递归地穷举所有可能的解，并在搜索过程中进行剪枝来减少搜索空间。回溯法通常用于求解满足约束条件的所有解或满足某个条件的一个解。

分支限界法是一种广度优先搜索算法，通过对搜索空间进行剪枝和优先级排序，选择最有希望的分支进行扩展。分支限界法通常用于求解最优解或找到满足某个条件的一个最优解。

(2) 求解上述问题的约束函数和上界函数：

约束函数：对于该问题，约束函数是确保每辆货车上的货物总重量不超过其载重量的条件。即：

对第一辆货车，约束函数为：载重量  $\leq 70$

对第二辆货车，约束函数为：载重量  $\leq 80$

上界函数：上界函数用于估计当前节点的最优解上界，以指导搜索过程。对于该问题，上界函数可以是剩余货物的总重量，因为当前节点的最优解上界不能超过剩余货物的总重量。即：

上界函数 = 剩余货物总重量

(3) 子树测试尽快生效的操作是剪枝。剪枝是一种在搜索过程中判断某个节点是否需要继续扩展的策略，可以提前舍弃不符合要求的节点，减少搜索空间。

为使子树测试尽快生效，可以采用以下操作：

引入合适的启发式策略：选择合适的启发式策略来优先扩展那些有可能更接近最优解的节点，从而更早地剪枝无效的节点。

剪枝策略：利用约束条件、上界函数和问题特性设计剪枝策略，尽早剪去不符合要求或不可能成为最优解的节点，减少搜索空间。例如，在载重量已经超过了货车载重量的节点上进行剪枝。

通过合理的剪枝操作，可以减少不必要的搜索，提高算法的效率。