

主要考查内容

- 算法、时间复杂度、空间复杂度的概念
- 排序算法的复杂度，平均、最好、最差
- 贪心、分治、回溯、递归、动态规划、线性规划
- 树、图
- P、NP、NPC、NP-Hard

选择题（10 题，每题3 分，共30 分）

选择题给出两道示例，考试有 10 个题。

1. 使用（ ）策略的算法不一定能得到最优解。

(A) 蛮力法 (B) 贪心法 (C) 动态规划法 (D) 回溯法

2. 使用分治法求解不需要满足的条件是（ ）。

(A) 子问题必须是一样的 (B) 子问题不能够重复 (C) 子问题的解可以合并 (D) 原问题和子问题使用相同的方法解

算法题（6 题，每题10 分，共60 分）

考核方式：以下三种方式选一种

- 写一个类或者函数
- 使用文字写出算法主要流程、思想
- 使用伪代码表达

最主要的是你关于这道题的思考。

1 数组中的第k个最大元素

在未排序的数组中找到第 k 个最大的元素。请注意，你需要找的是数组排序后的第 k 个最大的元素，而不是第 k 个不同的元素。

说明:你可以假设 k 总是有效的，且 $1 \leq k \leq$ 数组的长度。

示例 1:

输入: [3,2,1,5,6,4] 和 $k = 2$

输出: 5

示例 2:

输入: [3,2,3,1,2,4,5,5,6] 和 $k = 4$

输出: 4

2 漂亮数组 分治

对于某些固定的 N , 如果数组 A 是整数 $1, 2, \dots, N$ 组成的排列, 使得:

对于每个 $i < j$, 都不存在 k 满足 $i < k < j$ 使得 $A[k] * 2 = A[i] + A[j]$ 。

那么数组 A 是漂亮数组。

给定 N , 返回任意漂亮数组 A (保证存在一个)。

示例 1:

输入: 4

输出: [2,1,4,3]

示例 2:

输入: 5

输出: [3,1,2,5,4]

3 分发饼干 贪心

假设你是一位很棒的家长, 想要给你的孩子们一些小饼干。但是, 每个孩子最多只能给一块饼干。对每个孩子 i , 都有一个胃口值 g_i , 这是能让孩子们满足胃口的饼干的最小尺寸; 并且每块饼干 j , 都有一个尺寸 s_j 。如果 $s_j \geq g_i$, 我们可以将这个饼干 j 分配给孩子 i , 这个孩子会得到满足。你的目标是尽可能满足越多数量的孩子, 并输出这个最大数值。

注意: 你可以假设胃口值为正。一个小朋友最多只能拥有一块饼干。

示例 1:

输入: [1,2,3], [1,1]

输出: 1

解释: 你有三个孩子和两块小饼干, 3个孩子的胃口值分别是: 1,2,3。虽然你有两块小饼干, 由于他们的尺寸都是1, 你只能让胃口值是1的孩子满足。所以你应该输出1。

示例 2:

输入: [1,2], [1,2,3]

输出: 2

解释: 你有两个孩子和三块小饼干, 2个孩子的胃口值分别是1,2。你拥有的饼干数量和尺寸都足以让所有孩子满足。所以你应该输出2。

4 最长公共子序列 动态规划

给定两个字符串 text1 和 text2, 返回这两个字符串的最长公共子序列的长度。

一个字符串的 子序列 是指这样一个新的字符串: 它是由原字符串在不改变字符的相对顺序的情况下删除某些字符 (也可以不删除任何字符) 后组成的新字符串。例如, "ace" 是 "abcde" 的子序列, 但 "aec" 不是 "abcde" 的子序列。两个字符串的「公共子序列」是这两个字符串所共同拥有的子序列。

若这两个字符串没有公共子序列, 则返回 0。

示例 1:

输入: text1 = "abcde", text2 = "ace"

输出: 3

解释: 最长公共子序列是 "ace", 它的长度为 3。

示例 2:

输入: text1 = "abc", text2 = "abc"

输出: 3

解释: 最长公共子序列是 "abc", 它的长度为 3。

示例 3:

输入: text1 = "abc", text2 = "def"

输出: 0

解释: 两个字符串没有公共子序列, 返回 0。

5 复原IP地址 回溯

给定一个只包含数字的字符串，复原它并返回所有可能的 IP 地址格式。

有效的 IP 地址 正好由四个整数（每个整数位于 0 到 255 之间组成，且不能含有前导 0），整数之间用 '.' 分隔。

例如："0.1.2.201" 和 "192.168.1.1" 是 有效的 IP 地址，但是 "0.011.255.245"、"192.168.1.312" 和 "192.168@1.1" 是 无效的 IP 地址。

示例 1：

输入：s = "25525511135"

输出：["255.255.11.135", "255.255.111.35"]

示例 2：

输入：s = "0000"

输出：["0.0.0.0"]

示例 3：

输入：s = "010010"

输出：["0.10.0.10", "0.100.1.0"]

示例 4：

输入：s = "101023"

输出：["1.0.10.23", "1.0.102.3", "10.1.0.23", "10.10.2.3", "101.0.2.3"]

6 修车 线性规划

同一时刻有 N 位车主带着他们的爱车来到了汽车维修中心。

维修中心共有 M 位技术人员，不同的技术人员对不同的车进行维修所用的时间是不同的。

现在需要安排这 M 位技术人员所维修的车及顺序，使得顾客平均等待的时间最小。

说明：顾客的等待时间是指从他把车送至维修中心到维修完毕所用的时间。

输入格式

第一行有两个数 M,N，表示技术人员数与顾客数。

接下来 n 行，每行 M 个整数。第 i+1 行第 j 个数表示第 j 位技术人员维修第 i 辆车需要用的时间 $T_{i,j}$ 。

输出格式

最小平均等待时间，答案精确到小数点后 22 位。

示例：

输入：

2 2

3 2

1 4

输出：

1.50

简答题（1 题，10 分）

简单描述 P 问题，NP 问题，NPC 问题和 NP-hard 问题，以及他们之间的区别和联系。