

第八次作业

Lecturer: 杨启哲

Last modified: 2025 年 11 月 18 日

截止日期 2025 年 11 月 24 日晚 24: 00

重要提醒： 请邮件提交的同学严格按照第一个课件上的要求，平时作业请在邮件提交一个名为“**学号-姓名-算法第 8 次平时作业**”的 pdf 文件作为你的作业，编程作业请在邮件提交一个名为“**学号-姓名-算法第 8 次编程作业**”的 pdf 文件作为你的作业，邮件标题也请写成“**学号-姓名-算法第 8 次平时（编程）作业提交**”。

1. 对于任意给定的正整数 M , 请给出三个矩阵相乘的例子，它们的一种乘法顺序所需要的数量乘法的次数至少是另一种顺序的 M 倍。
2. 回顾贪心算法中我们讲到的货币兑换问题。有一个货币系统，它有 n 种硬币，面值分别为 $v_1 = 1, v_2, \dots, v_n$ 。现在需要兑换价值为 y 的钱，使得硬币的数量最少。之前提到过这个问题贪心算法不一定能给出最优解，现在请设计一个高效的算法解决这个问题。
3. 假设你现在有 n 枚硬币，其正面朝上的概率分别为 $p_1, p_2, \dots, p_n \in [0, 1]$ 。请设计一个 $O(n^2)$ 时间的算法，计算正面朝上的硬币数恰好为 k 的概率（硬币之间是相互独立的），这里假设对两个 $[0, 1]$ 区间内的数进行加法和乘法的运算时间为 $O(1)$ 。
4. 为了降低背包问题运行的时间界限 $O(nW)$ ，我们尝试用这样的思路，我们使用一个大数 K 将容量 W 和每件物品的体积 w_i 都缩小 K 倍。具体来说，令 $W \leftarrow \lceil \frac{W}{K} \rceil$, $w_i \leftarrow \lceil \frac{w_i}{K} \rceil$ ；再在这个背包问题的新实例上运行我们提出的解决背包问题的算法。
 - (1) 现在算法的时间复杂度是多少？
 - (2) 这样子的策略其实并不能总得到原实例的最优解，请给出一个反例。
5. 这个问题是对贪心算法解决背包问题的一个探索：
 - (1) 请给出一个反例，说明按照价值排序然后贪心选择并不能总得到最优解。
 - (2) 请给出一个反例，说明按照单位重量价值排序然后贪心选择并不能总得到最优解，并且该解与最优解的比值可以任意大。