

本次作业提交截止时间：2023年9月17日23:59

！ 注意事项

- 每次作业请上传至对应的文件夹，比如第一次作业上传至文件夹homework1。
 - 该文件夹中包含两个文件夹：code和pdf。
 - 非代码请上传pdf文件至文件夹pdf中。非pdf（比如doc文件、md文件、tex文件）请转换成pdf。
 - C++代码请上传至文件夹code中。本课程只允许C++代码。
 - pdf文件和cpp文件都用自己的学号命名。比如1001.pdf和1001.cpp。
 - 涉及公式的作业，推荐使用markdown编辑器。
 - 涉及算法伪代码的作业，推荐使用在线 $LATEX$ 编辑器[Overleaf](#)完成，提交编译生成的pdf文档即可。如果需要在overleaf中输入中文并能正确编译，首先在文档中加入package: `\usepackage[UTF8]{ctex}`，然后修改设置：设置->修改Latex引擎->选择“XeLatex”。
-

1. 算法导论第三版P35思考题 3-2

2. 算法导论第三版P35思考题 3-3

3. 算法导论第三版P35思考题 3-4

4. Let S be a set of n positive integers, where n is even.

1. Give an efficient algorithm to partition S into two subsets S_1 and S_2 of $n/2$ elements each with the property that the difference between the sum of the elements in S_1 and the sum of the elements in S_2 is **maximum**. What is the time complexity of your algorithm?
2. Suppose we change the word “maximum” to “minimum” in the above problem. Give an algorithm to solve the modified problem. What is the time complexity of your algorithm?

5. 你正在对各种型号的玻璃罐进行压力测试，以确定它们能从多高落下来而不会破碎。该实验的设置如下。你有一个 n 级的梯子，希望找到最高的梯级，可以在那里扔下一个罐子的样品，而不会破碎。该梯级称为该类罐子最高的安全梯级。

尝试二分搜索可能是很自然的：从中间梯级扔下一个罐子，看它是否破碎，然后根据结果尝试梯级 $n/4$ 或 $3n/4$ 。但这有一个缺点，就是会在寻找答案时打破很多罐子。

但是，如果你的主要目标是保留罐子，那么可以尝试以下策略。首先从第一个梯级扔下一个罐子，然后是第二个梯级，依次类推，每次爬一个梯级，直到罐子破碎。通过这种方式，你只需要一个罐子（在它破碎的那一刻，你有了正确答案），但你可能不得不扔它 n 次，而不是二分搜索时的 $\log n$ 次。

因此，这里有权衡：如果你愿意打破更多的罐子，那么似乎可以减少扔的次数。为了更好地在定量水平上理解这种权衡，考虑如何在 $k \geq 1$ 个罐子的固定预算下运行该实验。即你必须得到正确答案（最高的安全梯级），并且最多可以用 k 个罐子。

(a) 假定预算为 $k = 2$ 个罐子。描述一个寻找最高安全梯级的算法，要求你最多扔 $f(n)$ 次罐子，其中 $f(n)$ 是某个函数，比线性增长慢。换言之，应该有 $\lim_{n \rightarrow \infty} \frac{f(n)}{n} = 0$ 。

(b) 现在假设预算为 $k > 2$ 个罐子， k 是某个给定的常数。描述使用最多 k 个罐子找到最高安全梯级的算法。如果 $f_k(n)$ 表示该算法需要扔罐子的次数，那么函数 f_1, f_2, f_3, \dots 应该具有每个的增长渐近地慢于前一个的性质：即对于每个 k ，有 $\lim_{n \rightarrow \infty} \frac{f_k(n)}{f_{k-1}(n)} = 0$ 。