

本次作业提交截止时间：2023年10月2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 LAT_EX 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. 给你 n 根长度为整数的木棍，第 i 根的长度为 $L[i]$ 。现在要把它们切割成 k 根长度完全相同的木棍，当然可能会剩下一些碎片。出于某种考虑，希望这 k 根木棍尽可能的长。当 $L = [10, 6, 5, 3]$ ， $k = 4$ 时，显然你能切割得到4根长度为3的木棍。然而这并不是最佳方案，因为你能切割得到4根长度为5的木棍（想一想你该如何切割？）。给定 L 和 k ，设计一个高效的算法，返回从初始 n 根木棍切割出的 k 根相等木棍的最大可能长度。给出算法的基本思路和伪代码描述，分析算法的时间复杂度。
2. 在一个数组 $A[1..n]$ 中，**逆序对** (inversion) 是一对索引 (i, j) ，满足 $i < j$ 且 $A[i] > A[j]$ 。一个包含 n 个元素的数组中的逆序对数量介于0（如果数组已排序）和 $2n$ （如果数组完全逆序）之间。设计一个高效的算法计算数组 $A[1..n]$ 中逆序对的数量。给出算法的基本思路和伪代码描述，分析算法的时间复杂度。
3. 给定二维平面上两个不同的点 p 和 q ，如果 $p.x \leq q.x$ 且 $p.y \leq q.y$ ，称 q 支配 p 。给定一个点集 P ，设计一个高效的算法，计算每一个点 $p \in P$ 支配的点的数量。给出算法的基本思路和伪代码描述，分析算法的时间复杂度。
4. 算法导论第三版P124练习9.3-1
5. 基于下面的代码框架实现求解最近点对的分治算法。在100、1000、10000、100000个点上测试你的代码，将运行时间总结在提交的pdf文件中。一些代码注意事项如下：
 1. 代码中不要包含任何中文
 2. 不要使用<bits/stdc++.h>头文件
 3. 不要在main函数后放置任何代码
 4. 不要改变给定的函数原型

```
1 // Note:
2 // You are free to utilize any C++ standard library functions.
3 // Please ensure to include the necessary headers below.
```

```
4 // Avoid using <bits/stdc++.h> to prevent potential compilation errors that
   could result in a score of zero.
5 #include <iostream>
6 #include <complex>
7 #include <vector>
8 #include <utility>
9
10 using namespace std;
11
12 typedef long double LD;
13 typedef complex<LD> Point;
14
15 // You can add more functions here.
16
17
18 pair<Point, Point> closest_pair(vector<Point>& P)
19 {
20     // Insert you code here. You can add more functions but do not change the
   definition of this function.
21 }
22
23
24 int main(int argc, const char * argv[]) {
25     // You can insert code here to test you function.
26     return 0;
27 }
28
29 // Please refrain from including any code beyond the main function,
30 // as any additional code will be removed during the code evaluation process.
```