You are allowed to discuss with others but are not allowed to use any references other than the course notes and the three reference books. Please list your collaborators for each question. You must write your own solutions. See the course outline for the homework policy.

There are totally 52 marks. The full mark is 50. This homework is counted 10% of the course grade.

---

1. **Programming Problem:** $A + B = C$ **Problem** (20 marks)

   This question has two parts. The instructions for submitting your programs will be posted on piazza.

   (a) (10 marks) The first part is to write a program for fast polynomial multiplication, by extending the Karatsuba's $O(n^{1.59})$ algorithm for integer multiplication.

   Input: The first line has an integer n. The second line has $n+1$ integers $a_0, a_1, \ldots, a_n$, representing a degree n polynomial $f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$. The third line has $n+1$ integers $b_0, b_1, \ldots, b_n$, representing a degree n polynomial $g(x) = b_0 + b_1x + b_2x^2 + \ldots + b_nx^n$. You can assume that $0 \leq n \leq 100000$ and $|a_i| \leq 100000$ for $0 \leq i \leq n$ and $|b_j| \leq 100000$ for $0 \leq j \leq n$.

   Output: One line with $2n+1$ numbers, the coefficients of the degree $2n$ polynomial $f(x) \cdot g(x)$.

   Sample Input:
   ```
   5
   0 20 139 -78 137 -11
   -7 88 923 -342 179 0
   ```
   Sample Output:
   ```
   0 -140 787 31238 113634 -103819 177040 -70969 28285 -1969 0
   ```

   (b) (10 marks) The second part is to apply the first part to solve a counting problem.

   Given n positive integers $a_1, \ldots, a_n$, sorted in ascending order for convenience. Our goal is to count the number of ways to pick three of the numbers, so that the sum of two of them is equal to the third one. Specifically, count the number of triples of *indices* (i, j, k), such that $i < j < k$ and $a_i + a_j = a_k$.

   Input: The first line has a positive integer n. The second line has n positive integers $a_1, \ldots, a_n$. You can assume that $1 \leq n \leq 100000$ and $1 \leq a_1 \leq a_2 \leq \ldots \leq a_n \leq 100000$.

   Output: The number of triples of indices (i, j, k), such that $i < j < k$ and $a_i + a_j = a_k$.

   Sample Input:
   ```
   5
   1 1 2 2 3
   ```
   Sample Output:
   ```
   6
   ```
   Explanation: The triples of indices $(1, 2, 3)$ and $(1, 2, 4)$ correspond to the equation $1 + 1 = 2$. The triples of indices $(1, 3, 5)$, $(1, 4, 5)$, $(2, 3, 5)$, and $(2, 4, 5)$ correspond to the equation $1 + 2 = 3$. Note that we don't count the equation $2 + 1 = 3$ as the indices won't satisfy $i < j < k$. So the total number of triples is 6 (six).

2. **Written Problem: Solving Recurrences** (10 marks)

   Solve the following recurrence relations:

   (a) $T(n) = T(2n/3) + T(n/3) + n^2$.

   (b) $T(n) = \sqrt{n} \cdot T(\sqrt{n}) + n$.

   The base case is that constant size problems can be solved in constant time. Prove the best possible upper bound that you could for $T(n)$.

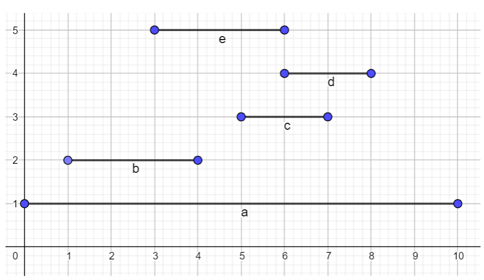
   You can either prove by induction or use the recursion tree method. Note that the master theorem as stated in L02 does not apply to these two problems.

3. **Written Problem: Visible Segments**

   (10 marks) In this problem, you are given as input a collection of horizontal line segments, and are asked to determine which segments are visible from above, and where. In particular, each segment has a height, h_i, and first and last x-coordinate, f_i and l_i. You may assume that all of the heights are distinct.

   For example, the following figure contains 5 segments, with data

name	h	f	l
a	1	0	10
b	2	1	4
c	3	5	7
d	4	6	8
e	5	3	6



   You are asked to divide the x-coordinate into intervals and report which segment is highest in each interval. For example, the answer for this example is

$$(0,1) : a \quad (1,3) : b \quad (3,6) : e \quad (6,8) : d \quad (8,10) : a$$

   Present a divide-and-conquer based algorithm that solves this problem in time $O(n \log n)$. Prove that your algorithm is correct and explain why it runs in the stated time. (There are other ways of solving this problem, but you will only get full credit for a divide-and-conquer solution.)

4. **Written Problem: Intersecting Squares**

(12 marks) You are given as input a collection of n unit squares in the plane. That is, each square has side length 1. Each square is specified by its lower, left corner. So, a square specified by (x, y) also has corners $(x + 1, y)$, $(x, y + 1)$, and $(x + 1, y + 1)$. Note that (x, y) may have non-integral values.

Your problem is to determine if this collection contains two squares that overlap, where we say that two squares overlap if there is some point that is strictly inside both of them. Touching at the boundaries doesn't count.

Design an algorithm that solves this problem in $O(n \log n)$ time. Prove that your algorithm is correct, and that it runs within the stated amount of time. (Substantial credit will be given for an algorithm that takes time $O(n \log^2 n)$ with correct proof.)