Course Name: Algorithm Design and Analysis

Modeling, and Divide And Conquer

For problems 1-6, you should do at least the following things:

1. Modeling: how you analyse the problem;
2. Algorithm description: describe your algorithm in **natural language**;
3. Time complexity: provide the time complexity and explain the reasoning behind it;
4. Space complexity: provide the space complexity and explain the reasoning behind it.

---

1. Longest Balanced Substring

   A string is called "balanced" if, for every letter that appears in the string, both its uppercase and lowercase forms are present at least once. For example, $aabAB$ is a balanced string, while abB is not. Given a string s, return the longest balanced substring of s.

2. Cutting Bamboo Poles

   A farmer has n bamboo poles of different lengths l_1, l_2, ..., l_n. He needs to cut these poles into m bamboo poles of the same length to build a raft. Find the maximum possible length of each bamboo pole for the raft.

3. Multiple Calculations

   Given a string s consisting of digits and operators $+$, $-$ and $*$, return all possible results from calculating the different possible ways to group the numbers and operators.

   Example:
   Input: s = "2*3-4*5"
   Output: -34, -14, -10, -10, 10
   Explanation:
   (2*(3-(4*5))) = -34
   ((2*3)-(4*5)) = -14
   ((2*(3-4))*5) = -10
   (2*((3-4)*5)) = -10
   (((2*3)-4)*5) = 10

4. N-sum

   There is an array $B[0..n-1]$ with n elements, where each element of B is an integer in $[0, n]$ (the elements are not necessarily different). You want to know if there exist n indices $i_1, i_2.....i_n$ (not necessarily different) such that

   $$\sum_{j=1}^{n} B[i_j] = m$$

   Where m is an integer in $[0, n^2]$.

   Design an $\mathcal{O}(n^2 \log n)$ time algorithm for this problem. You do not need to return the indices; just yes or no is enough.

   Hint: elements can be encoded using exponents!

5. Ex. Unary Cubic Equation

   Given a cubic equation of the form $ax^3 + bx^2 + cx + d = 0$, where the coefficients a, b, c, d are real numbers, determine the coefficients so that the equation has three distinct real roots (the roots should be in the range of -100 to 100), and the absolute difference between each pair of roots is $\geq$ 1.

   Output these three roots in ascending order on the same line (with a space between each root), and round each root to two decimal places.

   Hint: For the equation $f(x) = 0$, if there exist two numbers x_1 and x_2, where $x_1 < x_2$, and $f(x_1) * f(x_2) < 0$, then there must be a root between (x_1, x_2).

6. Ex. Distance

   You are given two integer arrays $arr1$ and $arr2$, and an integer d. Return the "distance value" between the two arrays.

   The "distance value" is defined as the number of elements that satisfy this distance requirement: for an element $arr1[i]$, there is no element $arr2[j]$ such that $|arr1[i] - arr2[j]| \leq d$.

   Example 1:
   Input:
   arr1 = [4, 5, 8],
   arr2 = [10, 9, 1, 8],
   d = 2

   Output: 2

   Explanation:

   For $arr1[0] = 4$, we have:
   $|4 - 10| = 6 > d = 2$
   $|4 - 9| = 5 > d = 2$
   $|4 - 1| = 3 > d = 2$
   $|4 - 8| = 4 > d = 2$
   So $arr1[0] = 4$ satisfies the distance requirement.

   For $arr1[1] = 5$, we have:
   $|5 - 10| = 5 > d = 2$
   $|5 - 9| = 4 > d = 2$
   $|5 - 1| = 4 > d = 2$
   $|5 - 8| = 3 > d = 2$
   So $arr1[1] = 5$ also satisfies the distance requirement.

   For $arr1[2] = 8$, we have:
   $|8 - 10| = 2 \leq d = 2$
   $|8 - 9| = 1 \leq d = 2$
   $|8 - 1| = 7 > d = 2$
   $|8 - 8| = 0 \leq d = 2$
   There are values that satisfy $|arr1[i] - arr2[j]| \leq 2$, so it does not meet the distance requirement.

   Thus, only $arr1[0] = 4$ and $arr1[1] = 5$ meet the distance requirement, and the distance value is 2.

   Use **divide-and-conquer** to solve this problem.