# Lab 4

1. **Missing Number in a Sequence:** Your input list of length $n$ consists of $n - 1$ consecutive non-negative integers in the range of 1 to $n + 1$, and one of the integers is missing in the list.

   **Example Test Cases:**

   $$\text{Input: } [1, 2, 3, 4, 6, 7, 8] \quad \text{Output: } 5$$

   $$\text{Input: } [2, 3, 4, 5, 6] \quad \text{Output: } 1$$

   $$\text{Input: } [1, 2, 3, 4, 5, 6, 7, 8, 10] \quad \text{Output: } 9$$

   —

2. **Maximum Product Subarray:** Given an integer array $A$, find a contiguous subarray that has the largest product, and return the product. Solve this using the divide-and-conquer technique.

   **Example Test Cases:**

   $$\text{Input: } A = [2, 3, -2, 4] \quad \text{Output: } 6$$

   $$\text{Input: } A = [-2, 0, -1] \quad \text{Output: } 0$$

   $$\text{Input: } A = [-2, 3, -4] \quad \text{Output: } 24$$

   —

3. **Majority Element Detection:** You are given an array of $n$ integers. A *majority element* is defined as an element that appears more than $n/2$ times in the array.

   **Example Test Cases:**

   $$\text{Input: } [3, 3, 4, 2, 3, 3, 5, 3] \quad \text{Output: } 3$$

   $$\text{Input: } [1, 2, 3, 4] \quad \text{Output: No Majority Element}$$

   $$\text{Input: } [2, 2, 1, 1, 1, 2, 2] \quad \text{Output: } 2$$

   —

4. **Deterministic Selection (Median of Medians):** Write a program to find the $k$-th smallest element in an unsorted array using the *Median of Medians* algorithm.

   **Example Test Cases:**

   $$\text{Input: } A = [7, 10, 4, 3, 20, 15],\ k = 3 \quad \text{Output: } 7$$

   $$\text{Input: } A = [7, 10, 4, 3, 20, 15],\ k = 4 \quad \text{Output: } 10$$

   $$\text{Input: } A = [12, 3, 5, 7, 19],\ k = 2 \quad \text{Output: } 5$$

—

5. **Karatsuba Multiplication:** Implement the Karatsuba algorithm for multiplying two large integers. Compare its performance experimentally with the traditional grade-school multiplication algorithm.

   **Example Test Cases:**

   $$\text{Input: } x = 1234,\ y = 5678 \quad \text{Output: } 7006652$$

   $$\text{Input: } x = 12345678,\ y = 87654321 \quad \text{Output: } 1082152022374638$$

   $$\text{Input: } x = 3141592653589793,\ y = 2718281828459045$$

   $$\text{Output:} 8539734222673565677848730527685$$

6. **k-closest element to the median:** Let $A$ be an array of $n$ distinct integers and let $k \leq n$ be a positive integer (note that $k$ may not be a constant). Design an algorithm, running in $O(n)$ time, that determines the $k$ numbers in $A$ that are closest to the median of $A$.

   **Example Test Cases:**

   (a) **Input:** $S = [2, 9, 1, 5, 7]$, $k = 3$ **Output:** $[5, 7, 2]$ (Median is 5, the three closest numbers are $5, 7$, and 2.)

   (b) **Input:** $S = [10, 3, 8, 15, 6, 20]$, $k = 2$ **Output:** $[8, 10]$ (Median is between 8 and 10, so the two closest numbers are 8 and 10.)

   (c) **Input:** $S = [1, 4, 9, 12, 20, 25, 30]$, $k = 4$ **Output:** $[9, 12, 20, 4]$ (Median is 12, the four closest numbers are $12, 9, 20, 4$.)

7. **Interval Overlapping:** Given a set of $n$ intervals $I = [a_1, b_1], [a_2, b_2], \ldots, [a_n, b_n]$, where $a_i < b_i$ for all $i = 1, 2, \ldots, n$. Devise a divide-and-conquer algorithm to compute the length of the biggest overlap between any two intervals in $O(n \log n)$ time. Justify the time complexity.

For example, the intervals $[1,7]$ and $[3,9]$ overlap, and the length of the overlap between them is

$$\min(7,9) - \max(1,3) + 1 = 4.$$

**Example Test Cases:**

(a) **Input:** $I = \{[1,7],[3,9],[10,15]\}$ **Output:** 4 (Biggest overlap is between $[1,7]$ and $[3,9]$, overlap length 4.)

(b) **Input:** $I = \{[2,5],[4,8],[9,12]\}$ **Output:** 2 (Biggest overlap is between $[2,5]$ and $[4,8]$, overlap length 2.)

(c) **Input:** $I = \{[1,3],[5,7],[9,11]\}$ **Output:** 0 (No intervals overlap, so maximum overlap length is 0.)

(d) **Input:** $I = \{[1,10],[2,6],[4,8]\}$ **Output:** 5 (Biggest overlap is between $[1,10]$ and $[4,8]$, overlap length 5.)