CMPSC 465 — Data Structures & Algorithms — **HW 2**

Fall 2024 — Mingfu Shao and Debarati Das

Due September 13th (Friday), 11:59 pm

Formatting: Each problem should begin on a new page. When submitting in Gradescope, try to assign pages to problems from the rubric as much as you can. Make sure you write all your group members' names. For the full policy on assignments, consult the syllabus.

1. **(0 pts.) Acknowledgements.** List any resources besides the course material that you consulted in order to solve the assignment problems. If you did not consult anything, write "I did not consult any non-class materials." The assignment will receive a 0 if this question is not answered.

2. **(20 pts.) Inversions.** Consider a company that maintains a large dataset of financial records, where each record is represented by an integer value. The company wants to identify pairs of records that meet specific comparative criteria for further analysis.

   You are given an array F of n integers, where each integer represents the value of a financial record. Your task is to efficiently identify and count the number of pairs of indices (i, j) in the array such that:

   - $1 \leq i < j \leq n$
   - $F[i] > 2 \times F[j]$

   - **(5 pts.) Simple Approach:** Describe a basic method to solve the problem by examining all possible pairs in the array.
   - **(10 pts.) Optimized Solution:** Develop an optimized algorithm that improves the efficiency of the initial approach. Explain your approach and provide pseudocode.
   - **(2 + 3 pts.) Time Complexity Analysis:** Evaluate the time complexity of both the simple and optimized approaches. Provide a detailed explanation of how you derive the complexity for each.

3. **(20 pts.) Median of Medians.** In the median of median algorithm, Anna and John decide to come up with their own variants. Anna decides to partition the original array into $\lceil n/7 \rceil$ groups of 7 elements each; whereas John decides to partition the original array into $\lceil n/3 \rceil$ groups of 3 elements each. The rest of the algorithm remains the same for both. You may assume that all entries in the original array are distinct.

   - **(7 pts.)** Analyze the bounds for $\max(|A_1|, |A_2|)$ and write the recurrence relation for the time complexity of Anna's algorithm in the following form:

   $$T(n) \leq \begin{cases} ?? & , n > n_0^A \\ \Theta(1) & , n \leq n_0^A \end{cases}$$

   Fill in the ?? in the above part. (You do not need to worry about a value for n_0^A.)

- **(7 pts.)** Analyze the bounds for $\max(|A_1|, |A_2|)$ and write the recurrence relation for the time complexity of John's algorithm in the following form:

$$T(n) \leq \begin{cases} ?? & , n > n_0^J \\ \Theta(1) & , n \leq n_0^J \end{cases}$$

  Fill in the ?? in the above part. (You do not need to worry about a value for n_0^J.)

- **(3 pts.)** Show that for Anna's algorithm, $T(n) = \Theta(n)$.
- **(3 pts.)** Show that for John's algorithm, $T(n) = O(n \log n)$.

Conclusion from question: In case you did not see it already, the above question implies that median of medians algorithm runs in linear time for partition-element sizes 5 and 7 (in fact for any odd number ≥ 5), but we can only prove superlinear times for size 3. We only consider odd partiton-elements as they have a unique median.