CS3230 Semester 2 2024/2025

Assignment 03
Correctness and Divide-and-Conquer

Due: Sunday, 16th February 2025, 11:59 pm SGT.

Instructions:

- Canvas Assignment Submission page: `Assignments/Assignment 3`.
- Please upload PDFs containing your solutions (hand-written & scanned, or typed) by the due date.
- Name the file **Assignment3_SID.pdf**, where SID should be replaced by your student ID.
- You may discuss the problems with your classmates at a high level only. You should write up your solutions on your own (any copying from your co-students or usage of Internet or AI tools is not allowed). Please note the names of your collaborators or any other sources in your submission; failure to do so would be considered plagiarism.
- Question listed as "graded for correctness" (worth 6 points) require complete answers. Other questions (worth 1 point each) will be graded only based on reasonable attempts. However, you should still do these questions, as they are practice questions, which would be useful for exams as well as for your knowledge.

1. (6 points; graded for correctness) You are consulting for the NUS Office of Student Affairs (OSA), which is investigating cheating cases involving bootleg, defaced student cards. The OSA has confiscated a collection of n student cards suspected of being unauthorized duplicates. Each student card is a small plastic card embedded with a chip that securely stores encrypted data. Each card's encrypted data correspond to a unique student (student number), and multiple student cards may correspond to the same student number.

   Since the cards are defaced, it is difficult to read the student number directly from the cards. Instead, OSA uses a 'matching device', a device capable of determining whether two student cards correspond to the same student number after performing some computations.

   The OSA poses the following question: among the collection of n confiscated cards, is there a subset of more than $n/2$ cards that all correspond to the same student number?

   Due to security concerns, you are not able to know the student number on a card, but are only allowed to interact with the matching device. Specifically, the only operation you can perform is to select two student cards and use the matching device to determine whether they correspond to the same student number.

   Your task is to design an efficient method to aid OSA's investigation using only $O(n \log n)$ queries to the matching device. Give arguments on why your algorithm works correctly, and analyze the query complexity of your algorithm (i.e., the number of queries made by your algorithm in terms of n).

2. (1 point) Suppose an array A is given. A pair (i, j) is called an *inversion* if $i < j$ and $A[i] > A[j]$. Design an algorithm to find the number of inversions in an array of size n. Give proof/arguments on why your algorithm is correct. Also, give the time complexity of your algorithm (provide and proof as tight a bound as possible).

3. (1 point) Consider the following algorithm for sorting:

   Input: Array $A[1..n]$ of integers.
   For $i = 1$ to $n - 1$ Do
   Invariant 1
   For $j = 1$ to $n - i$ Do
   Invariant 2
   If $A[j] > A[j + 1]$, then let $temp = A[j]; A[j] = A[j + 1]; A[j + 1] = temp$ Endif

    Endfor
Endfor
Output Array A.

We need to prove that the algorithm sorts the array.

a) Give an invariant for the array at the beginning of the first For loop (Invariant 1).

b) Give an invariant for the array at the beginning of the second For loop (Invariant 2).

c) Show that your invariants are true initially, are maintained by the algorithm, give the sorted array at the termination of the first For loop, and give the invariant for the next iteration of the first For loop at the termination of the second For loop.