

Assignment 7

csci2200, Algorithms

Instructions:

- HONOR CODE: WORK ON THIS ASSIGNMENT WITH AT MOST ONE PARTNER. BETWEEN DIFFERENT TEAMS, COLLABORATION IS AT LEVEL 1 [VERBAL COLLABORATION ONLY]
 - Write each problem on a separate page; If a problem has multiple parts, you can write all parts on the same page, as long as you leave space in between them.
-

1. **Majority element:** Suppose we are given an array A of length n with the promise that there exists a majority element (i.e. an element that appears $> \frac{n}{2}$ times). Additionally, we are only allowed to check whether two elements are equal (no $>$ or $<$ comparisons between the elements). Design an $O(n \lg n)$ algorithm to find the majority element, using divide-and-conquer. Explain the correctness and runtime of your algorithm.

[We expect: pseudocode, an English description of the main idea of the algorithm, an (informal) explanation of why it's correct; and running time analysis.]

2. **The inversion problem:** Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A . Give an algorithm that determines the number of inversions in an array in $O(n \lg n)$ time worst-case (Hint: modify merge sort).

We expect: pseudocode and a clear English description of what the algorithm is doing; A brief informal justification of why the algorithm is correct, and its runtime analysis.

Helper exercises (do not turn in).

- (a) List the inversions of the array $\langle 2, 3, 8, 6, 1 \rangle$.
- (b) What array with elements from the set $\{1, 2, \dots, n\}$ has the most inversions? How many does it have?
- (c) Give an algorithm that determines the number of inversions in an array in $O(n^2)$ time (this is the straightforward solution).