

15-451/651 Algorithm Design & Analysis, Fall 2026

Homework #1

Homework #1, Due: Wednesday, September 2nd at 11:59 pm

This homework has several short-answer problems followed by two written problems. The short-answer problems are graded on correctness only and do not require you to justify your answer or provide proofs unless asked for. Long-answer problems require you to justify your answers and provide proofs unless stated otherwise. On this homework, there is a SOLO problem and some GROUP problems. You may work in a group of up to 3 people on the GROUP problems. In all cases the write-up you submit must be written by you alone. You may not share your written solutions with each other. You can discuss the problem as a group, but then must write up your final solutions independently in your own words.

Submission: Solutions to the problems should be submitted as a PDF file using gradescope, with the answer to each problem starting on a new page. Each problem has a specified page limit that you should stick to. Your solutions must be typeset, not handwritten. You should use US letter-size paper, a font size no smaller than 10pt, margins no smaller than 1in. 10 points on this homework will be allocated to correctly following the style guidelines (see the Written HWs section of the [course policies](#)).

0. **Collaboration statement** Fill out the [collaboration form](#) to disclose your group members for this homework. If you worked solo, you must still fill out the form. Note that you will have to fill out this form separately for every written homework.
1. (SOLO, 15 pts) **Short-answer problems** (Limit: 1 page)
 - (a) **Academic Integrity:** (5 points) affirm that you have read the **course policy on academic integrity** and the **university policy on academic integrity** and that you agree to abide by them throughout the semester and beyond.
 - (b) **Power Ranks** (5 points) Describe an algorithm in the comparison model that given a list of n elements, returns a list containing the smallest, 3rd smallest, 9th smallest, 27th smallest, and so on, elements, i.e., the elements of rank 3^i for all i . Your algorithm should perform at most $O(n)$ comparisons.
 - (c) **Almost Sorting** (5 points) Suppose you are asked to "almost sort" an array of size n . An *almost sorted* array is sorted with the possible exception of the elements in positions 3, 4, and 5, which may appear in *any* order among those three positions. So for the sorted order [2, 4, 6, 8, 10], all of [2, 4, 6, 8, 10], [2, 4, 8, 6, 10], [2, 4, 10, 8, 6], etc. count as almost-sorted.
 - i. For $n \geq 5$ describe a maximal set of inputs to the problem such that any pair of them can not be almost-sorted by the same output (recall that the "output" for a sorting problem is a permutation of the input elements).
 - ii. Based on your answer to i, give the resulting information-theoretic lower bound on deterministic algorithms for this problem in the comparison model (no asymptotic notation).

2. (GROUP, 25 pts) **Shuffle** (Limit: 2 pages)

Supposed your friend has two decks of cards B and C , where each card is labeled with some integer. The decks are each sorted, but they may have different sizes and the cards within them may have arbitrary values. Your friend also is a huge fan of *bitonic sequences*, which is a sequence of numbers in which, the numbers are at first in increasing order, and after a certain point start decreasing. For example, $[1, 5, 7, 8, 4, 3, 2]$ is a bitonic sequence as it increases to 8 then decreases to 2. As such, your friend lays deck B down in increasing order and then places deck C on top in decreasing order, making a new deck A containing all n cards that is a bitonic sequence.

- (a) (5 Points) Given A , come up with an algorithm which uses $O(n)$ comparisons to split A into two sorted decks D and E which could be combined to form A the same way as B and C . As with all algorithms you present in this course, you should prove its correctness and complexity.
- (b) (10 Points) Given A , come up with an algorithm which uses $O(n)$ comparisons to sort A . Prove its correctness and complexity.
Hint: Use part (a).
- (c) (10 Points) Prove an $\Omega(n)$ lower bound on any deterministic algorithm sorting A in the comparison model.
Hint: Use an information-theoretic argument as discussed in lecture and recitation.

3. (GROUP, 25 pts) **Streak Sorting** (Limit: 2 pages)

In this problem, you are given an array of n distinct comparable elements $a_1, \dots, a_n$ in the comparison model. Define a *streak* to be a contiguous subsequence $a_i, a_{i+1}, \dots, a_j$, where $j > i$, such that the subsequence is increasing, namely, $a_i < a_{i+1} < \dots < a_j$. Suppose $a_1, \dots, a_n$ can be partitioned into t consecutive streaks, for some $1 \leq t \leq n/2$ that is unknown to you (i.e., your algorithm does not know t in advance).

- (a) Give an algorithm to sort the array $a_1, \dots, a_n$ using $O(n + n \log_2 t)$ comparisons. Briefly justify its correctness and the bound on the number of comparisons it performs
- (b) Show that for any deterministic algorithm solving this problem, there is at least one input which causes the algorithm to make $\Omega(n + n \log_2 t)$ comparisons.