

Assignment 4

csci2200, Algorithms

Instructions:

- HONOR CODE: WORK ON THIS ASSIGNMENT ALONE, OR WITH ONE PARTNER. BETWEEN DIFFERENT TEAMS, COLLABORATION IS AT LEVEL 1 [VERBAL COLLABORATION ONLY]
 - Check out the Homework guidelines on class website.
-

1. **Select the \sqrt{n} -closest:** Given an unordered sequence S of n elements (for simplicity, assume items are integers or real numbers), describe an efficient method for finding the $\lceil \sqrt{n} \rceil$ elements whose values are closest to (the value of) the median of S . What is the running time of your method? Aim for linear time.

What we expect: The rationale of the algorithm, pseudocode, analysis

2. **Merging sorted lists:** Assume you have k sorted arrays containing a total of n elements, and you want to merge them together in a single (sorted) array containing all n elements. For simplicity you may assume that the k arrays contain the same number of elements, namely n/k elements each.

- (a) Approach 1: merge array 1 with array 2, then merge the result with array 3, then merge the result with array 4, and so on. What is the worst-case running time ?

What we expect: Detailed analysis of this approach

- (b) Approach 2: split the set of k arrays into two sets of $k/2$ arrays, merge each one recursively, then use the standard 2-way merge procedure (from mergesort) to combine the two resulting arrays. What is the worst-case running time ?

What we expect: A recurrence , the recurrence depth, and the solution.

- (c) Approach 3: Give another approach (to merge the k arrays) that uses a heap, and runs in $O(n \lg k)$ -time.

What we expect: The idea of the algorithm, pseudo-code, analysis

Evaluation

This assignment (and all subsequent assignments) will be evaluated along four general criteria:

1. **Algorithm:** Is the algorithm clearly described ? Is the general idea included? Is high-level pseudocode included?¹
2. **Correctness:** Does the algorithm solve the problem?
3. **Analysis:** Is the running time of your algorithm analysed?
4. **Style:** Does it look professional and neat? Is the explanation written carefully in complete sentences, and well-organized logic? Is it easily human-readable? Is it complete yet concise? Is it easy to understand? These kinds of questions do not affect correctness but greatly affect how readable the algorithm is.

¹Pseudocode should be clear enough that a student who took 1101 can understand what your algorithm is doing and could implement it in a language of their choice, without thinking too hard. At the same time, pseudocode is **not actual code**, and should not include details that are straightforward and make the ideas hard to follow. For e.g. it is preferred to say “find the max element in the array” (basic straightforward process) rather than spell it out.