

CS251 - Data Structures and Algorithms

Fall 2024

PSO 4, Week 5

Question 1

(Heap sort) In the following questions, we consider Heap sort using **Heapify**.

- (1) Show the array $\{3, 4, 1, 0, 9, 2\}$ as it goes through Heap sort (in the ascending order).
 - (2) Given K number of sorted (ascending ordered) arrays each having N/K elements in it, your task is to merge all these arrays to form a N -element final sorted array (also in the ascending order).
- (2.1) Propose a simple solution to the problem which may run in $O(N \log(N))$ time.
 - (2.2) Can you propose a better algorithm to solve the problem? What is the time complexity of your proposed solution?

Question 2

(Merge sort) Merge sort is in its nature, a Divide-and-Conquer algorithm.

(1) Suppose that when doing a Mergesort you recursively break lists into 4 equal-sized sub-arrays instead of 2. Will you get a better runtime performance asymptotically?

(2) You are given two sorted arrays that are identical except that one of them is missing a single element. In other words, one array has length n and the other has length $n - 1$. The goal is to design an efficient algorithm with $O(\log n)$ runtime that finds the missing element.

Question 3**(Quick sort)**

- (1) Illustrate the operation of the **Partition** step in Quick sort on $A = [2, 8, 7, 1, 3, 5, 6, 4]$.
- (2) Can we understand the average-case runtime of Quick sort? What is the best policy for selecting the pivot value in the quick sort?

Question 4**(Counting sort)**

- (1) Illustrate the operations of Counting sort on $A = [6, 0, 2, 0, 1, 3, 4, 6, 1, 3, 2]$.
- (2) Describe an algorithm that, given n integers in the range 0 to k , preprocesses its input and then answers any query about how many of the n integers fall into a range $[a, b]$ (for some $0 \leq a \leq b \leq k$) in $\mathcal{O}(1)$ time. Your algorithm should use $\Theta(n + k)$ preprocessing time.