[Spring 2023] CSE 317: Design and Analysis of Algorithms [All sections]
Homework 1 | Due: February 17, 2023

Instructions: You need to submit it on `Gradescope`. Details for submission will shared soon on `LMS`. Direct your all queries to the course staff (refer to syllabus for contact information and office hours).

---

1. (10 points) Arrange the following functions according to their growth rates i.e., a function f stays before a function g if, $f = O(g)$. All logs are of base 2.

$10^{-6}n^7$	$\sqrt{n}(\log n)$	$n/\log n$	n^n	$2^{n^{2/3}}$	$(\sqrt{n})^n$	$\log\log n$	0.0003	$2^{n/\log n}$	$1/n^3$

2. (10 points) Given a sequence of integers $\boxed{1,\ 10,\ 12,\ 32,\ 57,\ 89,\ 122,\ 200}$, arranged in increasing order, construct a decision tree for binary search and compute the total number of comparison for $\boxed{x = 12}$.

3. (10 points) Sort the given sequence of integers $\boxed{92,\ -4,\ 14,\ -2,\ 32,\ 2,\ -78,\ 21}$, using MERGESORT.

4. (20 points) Let f and g be two asymptotically positive functions such that $f(n) = O(g(n))$. For each of the following statements, decide whether it is **true** or **false** and give a *proof* or a *counterexample*.
   1. $\log_a f(n) = O(\log_a g(n))$ for some $a > 0$,
   2. $2^{f(n)} = \Omega(2^{g(n)})$,
   3. $\sqrt{f(n)} = O(\sqrt{g(n)})$,
   4. $10^{100}f(n) = O(10^{-100} \cdot g(n))$,
   5. $2^{n/(1+\log f(n))} = O(2^{n/(1+\log g(n))})$.

5. (10 points) Consider an n-nodes complete binary tree, where $n = 2^d - 1$ for some d. Each node v of T is labeled with a distinct real number x_v (i.e., $\forall u, v \in T, u \neq v \Leftrightarrow x_u \neq x_v$). A node v of T is a *local minimum* if the label x_v is less than the label x_u for all vertices u that are joined to v by an edge.

   You are given such a complete binary T, but the labeling is only specified in the following implicit way: for each node v, yu can determine the value x_v by *probing* the vertex v. Show how to find a local minimum of T using only $O(\log n)$ *probes* to the vertices of T.

6. (10 points) Construct a decision tree with minimum depth for sorting 3 elements.

7. (15 points) We would like to choose between three divide-and-conquer algorithms for some problem. They are listed below in parts (a), (b), and (c). For each one of them, state the corresponding recurrence relation, and state a tight runtime bound.
   (a) If you solve 7 sub-problems of size $n/7$, then the cost of combining the solutions of the subproblems to obtain a solution for the original problem is $3n + 20$,
   (b) If you solve 16 subproblems of size $n/4$, then the cost for combining the solutions is 100,
   (c) If you solve 2 subproblems of size $n/2$, then the cost for combining the solutions is $5n^2 + 2n + 3$.

8. (15 points) Given an array A of n integers and an integer b. Design an efficient divide-and-conquer algorithms to compute the frequency of b in A (i.e., count the number of times b appears in A). Evaluate time complexity of your algorithm.
