Assignment

This is an **individual** homework assignment. All solutions have to be typed. No credit will be received for handwritten solutions (except for figures or long equations that might be handwritten). You need to submit your assignment on Canvas in the related digital repository by the deadline.

As it appears in the course syllabus, for the homework assignments, students are encouraged to discuss the problems with others, but you are expected to turn in the results of your own effort (not the results of a friend's efforts)". Even when not explicitly asked, you are supposed to justify your answers concisely.

Question 1)

(a) Given an integer list (an array) of size n, write an algorithm in **C++/Java** that adds up all values in the list and displays the result of the summation on screen.

(b) Count the number of basic operations in your algorithm. Explain and **show all your work**.

(c) Write the function f(n) expressing the time complexity of your algorithm correspondingly. Come up with the Big-O notation, then. Explain.

Question 2) Consider the array below:

24, 12, 38, 3, 45, 2, 56, 8, 100, 6.

a) Construct a binary tree out of the given array. Explain how you do so.

b) Build a Max heap out of the resulted binary tree. Show all your work step by step. In each step (i) draw the corresponding tree as well as (ii) the resulting array.

c) Analyze the algorithm you used to construct the heap in part b. Provide enough detail. Assume the array size is n.

Question 3) (a) Take the following list of functions and arrange them in ascending order of growth rate. That is, if function g(n) immediately follows function f(n) in your list, then it should be the case that f(n) is O(g(n)).

$$f1(n) = n$$
$$f2(n) = n^{10}$$
$$f3(n) = 2^n$$
$$f4(n) = 100^n$$
$$f5(n) = n \log n$$
$$f6(n) = n^2 \log n$$
$$f7(n) = n^n$$
$$f8(n) = n!$$

(b) In your arrangement in Q3 part (a), if you have fi(n) <=fj(n) that means fi(n)=O(fj(n)). Pick **each** of such **consecutive** pair of functions in your arrangement (let’s call them fi(n) and fj(n), where fi(n)<=fj(n) in the arrangement) and prove **formally** why you believe fi(n)=O(fj(n)).