Due Date: 9th Oct 2021
20% penalty for 1 day late
40% penalty for 2 days late
Submission not allowed afterwards

CS2009: Design and Analysis of Algorithms (Fall 2021)
Assignment 1
Total Marks: 100

---

1. Show the steps insertion sort uses to sort the following list of integers in the descending order (from the highest to the lowest / biggest to the smallest):

$$15, 6, 12, 8, 7, 1, 9, 3, 5, 23$$

   Show the value of the key variable, k, at each step. Explain briefly why time complexity of insertion sort is $O(n^2)$. Use Loop invariant to show its correctness. [5 Points]

2. Show the steps merge sort uses to sort the following list of integers in the descending order (from the highest to the lowest / biggest to the smallest): [5 points]

$$15, 6, 12, 8, 7, 1, 9, 3, 5, 23$$

   Consider the following variation on Merge Sort, that instead of dividing input in half at each step of Merge Sort, you divide into three part, sort each part, and finally combine all of them using a three-way merge subroutine. What is the overall asymptotic running time of this algorithm? [5 points]

3. Repeat for Quick Sort. Use Loop invariant to show its correctness. [5 points]

$$15, 6, 12, 8, 7, 1, 9, 3, 5, 23$$

4. Let suppose you are building an Emoji classification model. You find out that your model is able to classify one emoji type easily and face difficulty in classifying other emojis types. The one possible reason behind your model biasness toward one particular emoji could be due to class imbalance. Your training input may contain a lot of examples of one particular emoji and very few examples for other emojis.

   Your tasks is to determine whether there is a emoji type in input array $A = [a_1, a_2, a_3, \cdots, a_n]$ that appears more than n/2 times. The elements of the array are GIF files. However you can answer questions of the form: is $A[i] = A[j]$? in constant time.

   Design an $O(n \log_2 n)$ algorithm to solve this problem. [5 points]

   Design linear-time $O(n)$ algorithm for solving above problem [5 points]

code	emoji	label
:heart:	❤️	0
:baseball:	⚾	1
:smile:	😀	2
:disappointed:	😔	3
:fork_and_knife:	🍴	4

Figure 1: EMOJISET - a classification problem.

5. Let suppose you are given the task of creating two teams each of n players from the pool of 2n players for the competition. Each player has a numerical rating assigned to him/her according to their talent. Show plausible strategy for dividing players as fairly as possible to avoid talent imbalance between team 1 and team 2. Design an $O(n \log_2 n)$ algorithm to solve this problem. [10 points]

6. Prove $5n^2 - 2n + 30 = O(n^2)$. Determine the values of constant c and n_0. [5 Points] Prove $2n^2 \log_2 n + 3n^2 = O(n^2 \log_2 n)$. Determine the values of constant c and n_0. [5 Points]

7. Watch the video lecture on Big O, Big Ω and Big Θ notation from

   http://www.youtube.com/watch?v=6Ol2JbwoJp0. Write the summary of the lecture in your words. [10 Points]

8. Use Master Theorem, to calculate the time complexity of the following [15 points]

$$T(n) = 4T(\frac{n}{2}) + n^3. \tag{1}$$

$$T(n) = 3T(\frac{n}{2}) + n^2. \tag{2}$$

$$T(n) = 9T(\frac{n}{2}) + n. \tag{3}$$

9. Use Iteration Method, to calculate the time complexity of the following [10 points]

$$T(n) = 6T(\frac{n}{2}) + n, (T(1) = 1). \tag{4}$$

$$T(n) = 3 + (\frac{n}{2}), (T(1) = 0). \tag{5}$$

10. For each of the following questions, indicate whether it is T (True) or F (False) and justify using some examples e.g. assuming a function? [15 Points]

   - For all positive $f(n), g(n)$ and $h(n)$, if $f(n) = O(g(n))$ and $f(n) = \Omega(h(n))$, then $g(n) + h(n) = (f(n))$.
   - if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$, then we have $(f(n))^3 = \Theta((g(n))^3)$
   - if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$, then we have $(f(n))^2 = (g(n))^2$