**[CSL202]** **2024-25-M**

# HOMEWORK I

Due Date: **Aug 09, 2024**.

**Instructions.**

1. No need to submit the solutions.
2. Homework will be evaluated based on a short class quiz.

---

1. Define the following terms with an example (a) Time complexity of an algorithm (b) Worst case running time of an algorithm (c) Best case running time of an algorithm.
2. For the following pseudo-code, calculate the running time by drawing a table (consisting costs and number of times a line is executed) and finding an exact expression for the running time using the table. Finally compute the the value returned by the pseudo code (Give an exact expression).

```
SAMPLE(n)
s = 3
for i = 1 to n do
    for j = i to n do
        s = s^3
    end for
end for
return s
```

3. TRUE/FALSE: Justify your answer
   (a) $2n^2 + 3n + 1 = \Theta(n^2)$
   (b) $n^3 + n \log n + n = \Omega(n^2)$
   (c) $n \log n + \log n = \Theta(n)$
   (d) $2n^2 + 4 = o(n^2)$
   (e) $3n^3 + 7n^2 + 5n = o(n^4)$
4. Sort the following functions in increasing order of asymptotic (big-O) growth. If some have the same asymptotic growth, then indicate that.

   $1.01^n$, $4n$, $5^n$, $4 \log n$, $4n \log n$, $(\frac{3}{2})^n$, $n^{\log n}$, $\sqrt{n} \log n$, $n^2$, $2^{\log n}$, $n!$
5. Consider each of the following statements, assuming that all functions are non-negative. For each statement, if the statement is true then provide a proof from the scratch. If the statement is false then provide a counter example and demonstrate why the statement is false.
   (a) If $f_1(n) = \Theta(g(n))$ and $f_2(n) = \Theta(g(n))$, then $f_1(n) - f_2(n) = O(1)$.
   (b) If $f(n) = O(g(n))$ then $f(n) + g(n) = \Theta(g(n))$.

(c) If $f(n) = O(g(n))$ then $g(n) = \Omega(f(n))$.

(d) For any constant $b > 1$ the function $f(n) = 1 + b + b^2 + \ldots + b^n$ is in $\Theta(b^n)$.

6. Consider the function $h$ given by $h(n) = \log(n!) = \sum_{i=1}^{n} \log(i)$.

   (a) TRUE/FALSE: $h(n) = O(\log(n))$.

   (b) TRUE/FALSE: $h(n) = o(n^2)$

   Justify your answers.