Formative theoretical task

Department of Computer Science

ALGORITHMS & DATA STRUCTURES

Essential information:

- The **deadline** for submitting the formative assessment is the due date stated at the top of this page.
- This assessment is subject to the policies stated on the Formative Assessment Policies page. Please ensure that you have read and understood these policies before starting the assessment.

Read before starting:

1. Once you start the formative task, you must complete it within the allocated time. You cannot return to it once the allocated time runs out.
2. You must answer **all** questions.
3. This is an open book task, which means that you may access academic resources.
4. You are responsible for time-keeping during your exam attempt: we recommend that you use a third-party time-keeping device to manage your time.
5. We recommend using a computer and reliable internet connection.
6. The **deadline** for submitting the completed questions (including uploading any files) is the due date stated at the top of this page.
7. The exam submission point will remain available for 30 minutes after the deadline. After 30 minutes, incomplete attempts will be submitted, even if you have spent less time in the task than the time limit stated above.

Question 1

Group the following functions by their asymptotic complexity using Big-Oh notation. In addition, you must order them in increasing order of complexity.

$10 \log n$, $n^3 \log n + n^3$, $n^2 \log n + n^3$, $10\, n \log n$,

$n \log n^{20}$, 10, $n^3 \log n$, $n + 10$

Question 2

Suppose you have the following algorithm:

```
function two_loops(n: int)
    x = 1
    z = 0
    while x <= n do // Loop #1
      y = n
      while y > x do // Loop #2
        y = y - 1
        z = z + 1
      end while
      x = x + 1
    end while
```

Executing an assignment takes one unit of time, and all other operations take zero units of time.

1. What is the execution time of loop #2 in terms of x, y, n? You may assume that y is greater than or equal to x. Show the steps of your derivation.
2. What is the execution time of loop #1 in terms of x, y, n? Show the steps of your derivation.
3. What is the asymptotic complexity of the algorithm expressed as Big-Oh of the input size n?

Question 3

```
function compute_number(n):
    if n < 2 then
        return n
    end if
    left = 1
    right = n
    while left <= right do
        mid = left + (right - left) / 2 // uses integer division
        mid_squared = mid * mid

        if mid_squared == n then
            return mid
        end if
        if mid_squared < n then
            left = mid + 1
        else
            right = mid - 1
        endif
    end while
    return right
```

1. Show the working of the algorithm when n is equal to 64. You must show the state of the variables for each iteration in the format shown below.
   ```
   iteration 1: left = 1, right = 64, mid=1+63/2= 1+31=32,
                mid_squared=1024, left = 1, right=31.
   ```
   Note both `left` and `right` appear two times to display their states at the start and end of the loop.
2. What is the asymptotic complexity of the algorithm using Big-Oh notation? You must justify your answer.

Question 4

Consider the `main_algorithm` shown below that uses two processes `process_a` and `process_b` with time complexity $O(n^2)$ and $O(n \log n)$ respectively.

```
function process_a(array):
    // $O(n$^2$)$ complexity where n is the size of the array

function process_b(array):
    // $O(n \log n)$ complexity where n is the size of the array

function main_algorithm(array):
    Process_a(array)
    for k = 1 to n do
        process_b(array)
    end for
    return
```

What is the overall time complexity of the `main_algorithm` function? Justify your answer based on the complexities of `process_a` and `process_b`.