Homework 3 – Due Tuesday, June 11, 2023 09:00 pm

- Provide step-by-step explanations, not just answers. Answers without explanations will earn a small fraction of the points.
- Submit your solutions on Gradescope. Include information about your collaborators (or say "Collaborators: none").

Problems

1. (**Loop invariant, 5 points**) Consider the following pseudo-code:

```
# A is an array of size n
def mystery(int A[], int n)
    s = 0
    i = 1
    while (i <= n):
        s = s + A[i]
        i = i + 1
    return s
```

Select the **true** statements:

- ☐ Mystery adds up the elements from $A[1, ..., n]$
- ☐ The complexity of mystery function is $O(A)$
- ☐ The complexity of mystery function is $\Theta(n)$
- ☐ At the end of each while iteration, s has the sum of elements in $A[1, ..., i]$
- ☐ At the beginning of each while iteration, s has the sum of the elements in $A[1, ..., i-1]$

For the ones that you select as **true**, write a one sentence justification.

2. (**Algorithm analysis, 6 points**) Consider the following pseudo-code:

```
def enigma(int y, int z)
    x = 0
    while (z > 0):
        if (is_odd(z) == True):
            x = x + y
        y = 2*y
        z = z/2 # integer division (round-down)
    retorn x
```

(a) Use the following theorem to prove that enigma computes the product of y and z.

Theorem: For all natural numbers n, $2 \cdot \lfloor \frac{n}{2} \rfloor + (n \mod 2) = n$

(b) Give the complexity of enigma in terms of Θ notation.

(c) Elucidate the impact of varying values assigned to y and z on the performance of the enigma function, and present effective measures to alleviate this concern.

3. (**Functions hierarchy, 3 points**) Order the following functions by order of growth: i.e., put the functions from smallest to largest in terms of their "big O" notation. Put them on separate lines, if they are different, on the same line, if they are the same. List the asymptotic order of functions on each line. No further proof is needed.

   Here is a sample output, if the functions you want to order are $n^2+2, 3n^3, n+\log(n), 5n^3+n^2+2$, $n^2+\log(n)$ then the solution would be:

   1. $n+\log(n) = O(n)$

   2. $n^2+2, n^2+\log(n) = O(n^2)$

   3. $3n^3, 5n^3+n^2+2 = O(n^3)$

   Here are the functions (6 in total) that you have to order:

   $1.5^{1.5^n} \quad n^n \quad 3^{2n} \quad n^{\log n} \quad \sqrt{n} \quad 2^{n+1}$

4. (**Directed acyclic graph, 4 points**) A source vertex in a directed graph is one with an in-degree of zero. Prove that every directed acyclic graph has at least one source vertex.

5. (**Paths, 3 points**) Suppose that $\delta(v) \geq k$ for every vertex v of a graph G. Prove that the graph has a path of length at least k.

6. (**Car manufacturing, 9 points**) In a car factory, there are various assembly steps involved in manufacturing a car, and these steps often have dependencies on each other. You were hired to determine the optimal order of assembly steps to ensure a smooth production process. Let's consider the following assembly steps:

   - Engine Installation
   - Chassis Assembly
   - Electrical System Installation
   - Interior Installation
   - Exterior Paint
   - Final Inspection

   and the following dependencies:

   - Chassis Assembly depends on Engine Installation.
   - Electrical System Installation depends on Chassis Assembly.
   - Interior Installation depends on Electrical System Installation.
   - Exterior Paint depends on Chassis Assembly and Electrical System Installation.
   - Final Inspection depends on Interior Installation and Exterior Paint.

(a) Find how to represent this problem by using a directed graph. Make sure to give a precise description of how nodes, edges and directions are assigned.

(b) Given a new set of steps S and a new set of dependencies D, you want to find out whether it is even possible to manufacture a car. For this you need to know that there are no two steps that mutually depend on each other. (e.g. s_1 needs to be finished before s_2, but s_2 needs to be finished before s_1). Formulate what property your graph needs to have so that this doesn't happen. Give a one sentence explanation why.

(c) Assume you are given a directed graph G in an adjacency list format. You want to find all nodes that currently have *indegree* 0 and delete the nodes and their adjacent edges from G. Find a $\Theta(m + n)$ algorithm `UpdateGraphAndDegrees` to do this. Your algorithm takes the graph adjacency list as an input. It returns two objects; 1. the updated adjacency list of G after the deletions 2. an array `indegree` that contains the in-degree of node i in `indegree[i]`. The in-degree of the removed nodes in this array should be set to 0.

7. (**Programming Assignment, 10 points**) Login to Vjudge and submit your solution to the programming assignment.

   Solve all 4 problems involving sorting to get full points:

   - 3 out of 4 problems solved corresponds to 75% of the points.
   - 2 out of 4 problems solved corresponds to 50% of the points.
   - 1 out of 4 problems solved corresponds to 25% of the points.

   You can choose either Pypy 3 or Python 3 as your language. No other programming language will be accepted.