Student name: ______________________________ Student ID: ______________

Family name, Given name

School of Information and Communication Technology (ICT)

Data Structures and Algorithms

Writing time
120 minutes

Reading time
10 minutes

You do not need to write any code.

Submission consists of one file:

- A Word or PDF document
- You can draw/write on a paper, take the image, and insert it into your submission document.

Problem 1

```
function complexMystery(n):
    if n <= 1:
        return 1
    sum = 0
    for i from 1 to n:
        sum = sum + i
    return sum + complexMystery(n - 1)
```

(a) Manually compute complexMystery(4). Clearly show:

- Each recursive call
- The work done in the loop inside each call
- The return value at each step

(b) Derive a recurrence relation T(n) for the time complexity of the function, taking into account both the recursive call and the loop in each call. What is the Big-O time complexity?

Problem 2

A company has deployed n versions of its software, numbered from 1 to n. Unfortunately, one version introduced a bug, and all later versions are also buggy.

You are given access to a function isBadVersion(version) that returns:

- false if the version is good
- true if the version is bad

Your task is to find the **first** bad version using as few calls to isBadVersion(version) as possible.

Write a pseudocode solution and explain how it minimizes the number of checks.

Problem 3

You are given the list of keys:
[58, 16, 43, 27, 93, 34, 72]

and a hash function:
h(K) = K mod 7

(a) Construct the closed hash table using linear probing. Show each step of the insertion and explain how collisions are handled.

(b) What is the **maximum number** of comparisons required for a successful search?

(c) What is the **average number** of comparisons for a successful search? (Assume all keys are equally likely to be searched.)

Problem 4

Insert the following keys, in order, into an initially empty AVL tree:
[28, 64, 90, 11, 24, 88, 90, 55, 82]

Show the state of the tree after each insertion. Indicate where rotations occur and specify the type of rotation (single or double).

Problem 5

Consider the following directed, weighted graph represented as an adjacency list:

```
Graph G = {
  s: [(a, 4), (b, 2)],
  a: [(c, 3)],
  b: [(a, 1), (d, 5)],
  c: [(d, 1)],
  d: []
}
```

Use Dijkstra's algorithm to compute the **shortest distance** from vertex s to all other vertices.

Show:

- The state of the priority queue (or tentative distances) at each step
- The final distances
- The resulting shortest path tre

Problem 6

You are analyzing customer behavior at **Woolworths**, a major Australian supermarket. Each customer's journey is recorded as a sequence of aisle visits or actions. For example:

- **Customer A**: Fresh Produce → Dairy → Snacks → Frozen Foods → Checkout
- **Customer B**: Fresh Produce → Snacks → Frozen Foods → Bakery → Checkout

(a) Woolworths wants to understand how similar two customers' journeys are, even if they visited slightly different sections or skipped some. Propose a method to calculate the minimum number of operations (such as insertions, deletions, or replacements) required to transform one journey into the other. Clearly explain how your method works.

(b) Analysts have observed that the pattern pick item → browse → pick item often signals high engagement or indecision—possibly indicating interest in promotions or product comparisons. Detecting this behavior can help Woolworths optimize shelf layouts and deliver targeted promotions.

Given a long sequence of customer actions such as:
enter, pick item, browse, pick item, move, browse, pick item, browse, pick item, checkout

Your task is to describe an **efficient method** to detect how many times this specific pattern appears across all customer logs collected during the day. Explain how this approach is more efficient than checking each position manually.

Problem 7

Australia operates a nationwide **flood emergency response system** consisting of sensor stations located along rivers, dams, and flood-prone zones. Each station is identified by a unique number (e.g., 1, 2, 3, …). These stations are **interconnected** based on real-world geography and water flow—stations upstream may alert multiple downstream stations, and some stations may share communication links due to proximity.

The network of sensors does **not follow a regular structure** (such as a tree); instead, stations may have multiple connections, forming **complex and irregular topologies**. Some may even be connected in loops or have redundant backup paths.

When a **flood is detected** at a particular station:

- The alert must **spread to all reachable operational stations** as **quickly as possible** so that emergency actions can begin.
- Later, analysts may want to **trace how the alert reached a specific station**, possibly identifying **multiple potential paths** from the source.

However, some stations may be **offline** due to damage, which prevents the alert from passing through them.

Your Tasks:

1. **Describe an appropriate data structure** to model the sensor network. Justify your choice based on the network's characteristics and the operations that need to be performed.
2. Given the following sensor connections and assuming **station 4 is offline**, simulate:
    - The **order in which sensors receive the alert** if the flood is first detected at station 1.
    - The **possible paths** the alert could have taken to reach station 7.

Connections:

1 – 2, 3

2 – 4, 5

3 – 6

5 – 7

6 – 7