[VIT University] – Algorithm Design and Analysis

End Semester Examination (2025)

Total Marks: 50
Instructions:

- Each answer should contain the following components: **Logic, Illustration, Pseudocode, and Time Complexity Analysis.**
- Clearly specify and define any assumptions made.
- **DO NOT write actual code**—only provide the requested components.

---

1. (Greedy Algorithm) [15 Marks]

A company wants to assign its projects to a set of available employees to maximize the total number of projects that get completed. Each project requires a unique skill set, and each employee has a list of skills they possess. A worker can handle multiple projects if they possess the required skills.

Given:

- A set of projects **P** = $\{P_1, P_2, P_3, \ldots, P_k\}$
- A set of employees **E** = $\{E_1, E_2, \ldots, En\}$
- A function **skills(E_i)** that returns the set of skills an employee possesses.
- A function **req(P_i)** that returns the set of required skills for a project P_i.

Write an algorithm using the **greedy technique** to find a subset of employees **E' ⊆ E** such that all projects in **P** have at least one assigned employee and the total number of employees in **E** is minimized.

Your answer must include:

- **Logic** explaining the approach.
- **Illustration** with an example.
- **Pseudocode** to implement the algorithm.

- **Time Complexity Analysis.**

---

2. (Dynamic Programming) [10 Marks]

Given an array **A** of length **n**, a subarray is defined as any contiguous subarray of **A**. A mountain subarray is a contiguous subarray that first **increases**, then **decreases**. The goal is to **find the length of the longest mountain subarray** in a given integer array **A** of length **n**.

Definition:
A subarray **A[l], A[l+1], ..., A[r]** is a **mountain** if:

1. There exists an index **i** where **l < i < r** such that:
    - **A[l] < A[l+1] < ... < A[i]** (strictly increasing)
    - **A[i] > A[i+1] > ... > A[r]** (strictly decreasing)
2. If no mountain exists, return 0.

Example:
Input: **A = [2,1,4,7,3,2,5,3,1,0,4,6,2,1]**
Output: **5**
(One longest mountain subarray is [0,4,6,2,1], and another is [1,2,5,3,1])

Your answer must include:

- **Logic** explaining the approach.
- **Illustration** with an example.
- **Pseudocode** to implement the algorithm.
- **Time Complexity Analysis.**

---

3. (Backtracking) [15 Marks]

A **robotic cleaner** starts at a given position **S(x, y)** on a **m × n** floor grid. It can move **up, down, left, or right**, but cannot pass through obstacles. The goal is to check if the robot can reach a charging station **G(p, q)** following a sequence of valid moves while avoiding obstacles.

Given:

- A **m × n** grid with some blocked cells marked as **X**.
- The robotic cleaner starts at **S(x, y)** and can move up, down, left, or right.
- The goal is to determine whether there is a valid path from **S(x, y)** to **G(p, q)** without crossing an obstacle.

Write a recursive **backtracking algorithm** to check if the robotic cleaner can reach **G** from **S** while avoiding obstacles.

Your answer must include:

- **Logic** explaining the approach.
- **Illustration** with an example.
- **Pseudocode** to implement the algorithm.
- **Time Complexity Analysis.**

---

4. (String Processing) [10 Marks]

You are given a long string **T** of length **n** consisting of lowercase English letters. Your task is to find the **longest substring that appears at least twice** in T. The substring must be contiguous and non-overlapping.

Example:
Input: **T = "banana"**
Output: **"ana"**
Explanation: The longest repeated substring is **"ana"**, which appears twice.

Your answer must include:

- **Logic** explaining the approach.
- **Illustration** with an example.
- **Pseudocode** to implement the algorithm.
- **Time Complexity Analysis.**