NAME: TRISHA MAE A. REPOLONA & CHRISTINE JOY BANDE   SUBJECT: DATA STRUCTURE AND ALGORITHM

COURSE & YEAR: BSIS – 2A   PROFESSOR: MS. KHRISTINE BOTIN

Activity Title: Treasure Hunter's Heap Challenge

Theme: Organize your treasure collection by tracking the most and least valuable items using heaps!

Objective:

Engage students in an activity to learn about Heap Data Structures by simulating a treasure-hunting game. This activity would help the students utilize the knowledge they gained about Min-Heaps and Max-Heaps to manage and prioritize treasures in a proper manner and develop creativity, problem-solving, and critical thinking skills.

Learning Goals:

- Upon completion of this exercise students should be able to:
- Demonstrate the concept of a Max-Heaps and Min-Heaps by implementing an insertion.
- Use heaps to find maximum and minimum values without much difficulty.
- Apply heapify operations so that they maintain heap properties.

Instructions

1. Set the Scene: Imagine you are a treasure hunter with a room full of treasures. You need to sort treasures by their value, find the most and least valuable treasures, and approximate how heap operations would work in the real world.

    Examples:

    - Monitoring the most and least valuable treasures excavated from old ruins.
    - Controlling an inventory of gold coins in a pirate's treasure chest.
    - Sorting treasures based on value to present to a king or queen.

Tasks

- Insert Treasures: Write a function that takes in treasure values to be inserted into both Max-Heap and Min-Heap.
    a. Input: Treasure values, for example, 30, 50, 20.
    b. Output: Print out both heaps after each insertion.

- Find the Most and Least Valuable Treasures: Write programs to find the treasure worth most in the Max-Heap and the treasure worth least in the Min-Heap.
    a. Input: N/A.
    b. Output: Print max and min value from the respective heap

- Incorporate a Delay for Reality: Implement a delay of 2 seconds for the user to select another action as if he was really supposed to analyze treasures.

Instructions

1. Initialization:
    a. Develop a program for managing treasures based on heaps.
    b. Use two different arrays as follows for Max-Heap and Min-Heap.
    c. Include delays and themed print statements.

2. Input Examples:
    Treasures: 30, 50, 20

Output Example

Max-Heap: [50, 30, 20].

Min-Heap: [20, 30, 50].

Most valuable treasure: 50.

Least precious treasure: 20.

3. Program Flow: Present a menu with actions:
    - Add treasures.
    - Show the most valuable treasure.
    - Show the least valuable treasure.
    - Exit.