Algorithms Project

Task1

a. **Write all required algorithms to sort a sequence of numbers using Heap-Sort:**
algorithm 1: Max-Heapify
algorithm 2: build max-heap
algorithm 3: heap-sort

b. **Analyze in detail your written algorithms in Part (a).**

1. Max- Heapify Time Complexity
    - The heapify function runs in O(log n) because it traverses down a binary heap structure.
2. Build Max-Heap Time Complexity
    - The build_max_heap runs in O(n). This is because it calls heapify for all non-leaf nodes starting from the last non-leaf node toward the root.
3. Heap-Sort Time Complexity
    - The main loop in the Heap-Sort runs for n-1 iterations, and for each iteration, the heapify function (running in O(log n)) is called. Therefore:

      Time complexity: O(n log n)

Task2

a. **Write all required algorithms to find MST using Kruskal's Algorithm**

- Sort all edges in the graph in non-decreasing order based on their weights.
- pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. if a cycle is not formed, include this edge. else, discard it.
- repeat step 2 until there are (V-1) edges in the spanning tree.

b. **Analyze in Detail the Written Algorithms**
sorting ⇒ O(ElogE)
find , union ⇒ O(ElogV)