Week 10 - Assignment (3 marks)

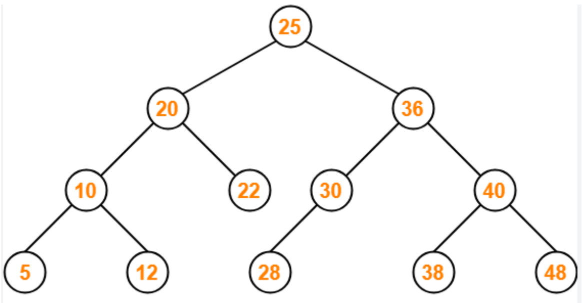
Note: Try to make the best use of appropriate C++ features.

Extend the provided Treap class with additional functionality to deepen your understanding of Treap operations and the use of augmented data structures in binary search trees.

Tasks:

You are provided with a basic implementation of a Treap class that supports fundamental operations such as insertion, deletion, and search. Your task is to extend this implementation by augmenting each node with additional information and modifying relevant functions accordingly.

1. **Subtree Key Sum:** Modify each node to maintain the **sum** of all keys in its subtree, **including itself**. For example, if node 20 has subtree nodes 10, 22, 5, and 12, then the sum stored at node 20 should be 69 (= 20 + 10 + 22 + 5 + 12).



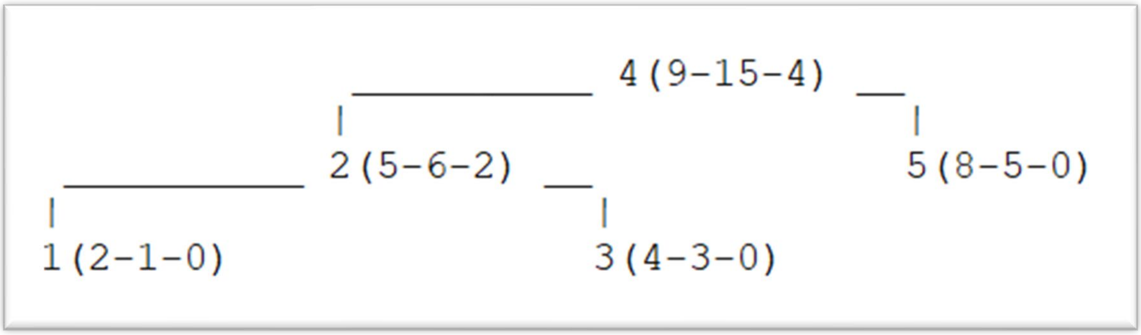
2. **Subtree Node Count (Excluding Self):** Add a field to each node to track the **count** of nodes in its subtree, **excluding the node itself**.
3. **Modified Output Function:** Update the print-related functions to produce the required **output** below. For each node, it should display the format: **key (priority-sum-count)**

Input:

Please use ***the provided run_wa10.cpp*** to complete these tasks.

Don't modify the main function!!!

Ouput: (should be like the below) **key** (**priority-sum-count**)



```
                   _____________  4(9-15-4)  __
                  |                            |
 ______________  2(5-6-2)  __                  5(8-5-0)
|                           |
1(2-1-0)                    3(4-3-0)
```

Submit:

1, **all C++ source code**

2, **wa10.txt**: a txt file contains all the source code for *plagiarism review*.

3, **output.jpg** (or png, bmp): a screenshot of the output by your program.

Please refer to the submission page for the Marking Rubric.