Lab 3:

Write a program and also find their amortized cost to implement B-tree for t=2 and 3

When t=2:

Output:

Enter the number of keys: 5

Enter the keys: 10 20 5 30 15

Amortized cost per insertion: 2.60

Aggregate cost: 13

Potential method cost: 15

Accounting method cost: 13

Insertion operations: 5

Split operations: 2

When t=3:

Enter the number of keys: 5

Enter the keys: 10 20 5 30 15

Amortized cost per insertion: 2.20

Aggregate cost: 11

Potential method cost: 15

Accounting method cost: 11

Insertion operations: 5

Split operations: 1

Analysis:

1. B-Tree Structure: The code defines a B-tree node structure and functions for inserting keys into the tree.

2. Time Complexity: Inserting N keys into the B-tree has a time complexity of O(N * log_base_T(N)), where T is the B-tree order.

3. Cost Metrics:
   - Total Cost (`totalCost`): The sum of insertion and split costs.
   - Amortized Cost (`amortizedCost`): Average cost per insertion.
   - Aggregate Cost (`aggregateCost`): Total cost.
   - Potential Method Cost (`potentialCost`): An alternative cost model.
   - Accounting Method Cost (`accountingCost`): Includes node credits.

4. Printed Metrics: The code calculates and prints these cost metrics along with insertion and split counters.
This analysis helps evaluate the efficiency of B-tree insertions using different cost models.

Calculation Part:

1. Total Cost (totalCost): The sum of all insertion and split costs.
   - Formula: Total Cost = (insertionCounter * insertionCost) + (splitCounter * splitCost)

2. Amortized Cost (amortizedCost): The average cost for each insertion, calculated by dividing the total cost by the number of insertions.
   - Formula: Amortized Cost = totalCost / insertionCounter

3. Aggregate Cost (aggregateCost): The overall cost, including both insertion and split costs.
   - Formula: Aggregate Cost = totalCost

4. Potential Method Cost (potentialCost): An alternative cost calculation assuming that each split operation stores T-1 credits for future insertions.
   - Formula: Potential Method Cost = (insertionCounter * insertionCost) + (splitCounter * splitCost * (T - 1))

5. Accounting Method Cost (accountingCost): This considers credits assigned to nodes during splits and adds the root's credits to the total cost.
   - Formula: Accounting Method Cost = totalCost + root->credits

These cost metrics help assess the efficiency of B-tree insertion operations and offer different perspectives on the costs involved.
Time complexity:

- Best Case: Inserting a key into a balanced B-tree with no splits takes O(log_base_T(N)) time.
- Worst Case: When every insertion triggers splits up to the root, it takes O(N * log_base_T(N)) time.
- Average Case: On average, inserting keys takes O(log_base_T(N)) time, assuming the tree remains reasonably balanced. Precise analysis considers various insertion patterns and tree structures using probabilistic methods.