Worst-Case Analysis of Inserting N Keys Using Linear Probing with Array Resizing

In the **worst case**, inserting N keys into an initially empty hash table using *linear probing* with *array resizing* can require a significant number of comparisons. Let's analyze this situation.

Understanding Linear Probing with Array Resizing

1. **Linear Probing**: In linear probing, if a collision occurs (i.e., the position determined by the hash function is already occupied), we check the next position in the array (wrapping around to the beginning if necessary) until we find an empty slot. Each unsuccessful check counts as a comparison.

2. **Array Resizing**: To ensure that the hash table does not become too full (which would degrade performance), we typically resize the array when the load factor (the ratio of the number of keys to the size of the array) exceeds a certain threshold (often 0.5). Resizing usually involves doubling the size of the array and rehashing all the existing keys.

Analysis of the Worst-Case Comparisons

1. **Worst-Case Scenario for Linear Probing**: In the worst case, inserting a new key may require probing many consecutive positions in the array before finding an empty slot. This can occur if there are clusters of keys, resulting in many comparisons to resolve each collision.

2. **Array Resizing Cost**:

   - Every time the array size doubles, all existing keys must be rehashed and reinserted into the new array. This rehashing step incurs additional comparisons.

Total Number of Comparisons in the Worst Case

1. **Initial Insertion Costs**: If the hash table is resized multiple times as it grows, the cost of inserting keys grows cumulatively.

2. **Cost of Resizing and Rehashing**:

   - When the table size doubles, we rehash all existing keys. Since the table size starts at a small number (e.g., 1 or 2) and doubles repeatedly, the total cost of rehashing can be expressed as:

$$\text{Total cost of rehashing} = N + \frac{N}{2} + \frac{N}{4} + \cdots + 1$$

- This series sums to $O(N)$ for rehashing operations.

3. **Cost of Linear Probing During Insertions**: The worst-case cost for probing can be as high as $O(N^2)$ comparisons in extremely unfavorable cases, such as when there is severe clustering.

Worst-Case Complexity

Combining both the cost of rehashing and the cost of linear probing:

$$\text{Total comparisons in the worst case} = O(N^2)$$

Conclusion: In the *worst case*, inserting N keys into a hash table using linear probing with array resizing can take up to $O(N^2)$ comparisons due to the potential clustering effect and the cumulative cost of probing. The resizing and rehashing operations add an $O(N)$ cost, but the primary contributor to the worst-case scenario is the linear probing cost.