1. Suppose we have a stack with operations PUSH, POP, and MULTI-POP. Any sequence of \( n \) operations has complexity \( O(n) \) for this data structure. Explain in your own words whether the addition of the operation MULTI-PUSH would change the analysis in any way. Justify your answer.

2. Suppose you have a linked list with the following operation:

\[
\text{op}(LL, n) = \begin{cases} 
\text{add } n \text{ to } LL, \text{ then add } n \text{ to } LL \text{ a second time } & \text{, if size is even} \\
\text{add } n \text{ to } LL, \text{ then double each element in } LL & \text{, if size is odd}
\end{cases}
\]

Doubling each element in \( 1\rightarrow2\rightarrow3 \) leads to \( 1\rightarrow1\rightarrow2\rightarrow2\rightarrow3\rightarrow3 \). Show the following:

(a) What is the worst case complexity of this operation?
(b) What is the worst case complexity of a sequence of \( n \) operations?

3. Suppose you have an array. The array has the following operation:

\[
\text{reduce}(A) = \begin{cases} 
\text{remove } k_i \text{ elements in } A. \ k_i = \frac{k_{i-1}}{2}
\end{cases}
\]

Each time we call \text{reduce} the number of elements removed is different. The first time called it removes \( k_0 \) elements. Suppose the array has enough elements so it is never empty.

(a) Suppose \( k_0 = 16 \). Write the summation for the cost of a sequence of 4 \text{reduce} operations.
(b) What is the complexity of \( n \) operations as a function of \( k_0 \)?
(c) If we change how we update \( k_i \) to:

\[
k_i = \frac{k_0}{i}
\]

What is the complexity of \( n \) operations as a function of \( k_0 \)?

4. Consider a dynamic table that triples its size whenever it becomes full. Use the potential method to show the amortized complexity of the insertion operation. Be sure to write the potential function used.

5. Insert the following keys in a B-tree of order 3 in the order given: 3, 23, 1, 2, 54, 9, 10, 6, 18, 11. Show all intermediate steps. After all keys are inserted, delete the following keys one by one in this order: 54, 23, 11, 18, 10, 2, 1. Show all intermediate steps. Follow the procedures exactly as given in the textbook. You must draw the trees yourself, no printouts will be accepted for this problem.