Question 3: Show O() by mathematical analysis - Show all work/algebraic summative derivations.

Given: Simple Vector classes, Optimized Simple Vector variations, and Simple Vectors with Linked Lists.

Simple and Optimized Simple Vector Approach: For the given vectors, the math reflects the actual operations of the class. Let's consider the access, push back, inserts, pop back, and delete functions of the vectors.

Push Back : T(n) = C + rSize * C2 + $\sum_{i=0}^{n-1} 1$, where C is a typical insert, and the summation represents copying of n elements into a new, bigger array. This analysis yields O(1) after summation T(n) = C + (1/n) * (C2 + n) = O(1).

Push Front : Inserting at the beginning requires pushing all existing elements, which means iterating through all n elements of the vector. Meaning, T(n) is C + $\sum_{i=0}^{n-1} 1$, which can simplify to C + n, equaling O(n).

Simple Vector with Linked List Approach:

Push Back : T(n) = C1 + $\sum_{i=0}^{n=1} C2$, where C1 is the head and the summation is the traversal to the end of the list and then add the node. The summation simplifies to C1 + C2 * n which equals O(n) timing.

Push Front : T(n) = C because it's a constant time operation regardless of list size. Meaning a new node HAS to be created if Push wants to be executed, along with pointer adjusting. So the complexity is O(1).