Insertions: Mathematical Analysis

Simple Vector Using Arrays

- Each insertion copies all previous elements into a new array of size (n + 1)
- Total Time for n insertions: 1 + 2 + 3 + ... + n = $\frac{n(n+1)}{2}$
- Average Time per insertion: $\frac{1}{n} * \frac{n(n+1)}{2} = \frac{(n+1)}{2}$ = O(n)

Proof: $\frac{n+1}{2} \leq c * n$ where c is a constant

$= \frac{n+1}{2N} \leq c = \frac{1+\frac{1}{N}}{2} \leq c$

As N approaches ∞, $\frac{1}{N}$ approaches 0, so the left-hand side approaches $\frac{1}{2}$

So, for all n ≥ 1, the left-hand side is always less than 1.

so: $\frac{n+1}{2} \leq n$ for all n ≥ 1 $\therefore \frac{\mathbf{n+1}}{\mathbf{2}}$ **= O(n)**

Optimized Simple Vector using arrays

- Array capacity doubles when full.
- all copies for n insertions: 1 + 2 + 4 + 8 + ... + $\frac{n}{2}$ = geometric series = 2n - 1 = O(n)
- Average per insertion: $\frac{1}{n} * \mathrm{O}(n) = \frac{O(n)}{n}$ = O(1)

Simple Vector Using Linked List

- Each insertion adds a node to the end
- Total Time for n insertions: n * O(1) = O(n)