Section 2: Dynamic Arrays and Amortized Analysis (2 questions)

Question 1

1/1 point (graded)

Let's imagine we add support to our dynamic array for a new operation PopBack (which removes the last element). PopBack will reallocate the dynamically-allocated array if the size is ≤ the capacity / 2 to a new array of half the capacity. So, for example, if, before a PopBack the size were 5 and the capacity were 8, then after the PopBack, the size would be 4 and the capacity would be 4.

What is the worst-case time for any n operations starting with an empty array?
Recall that the PushBack operation in a standard dynamic array increases its capacity twice when its size exceeds capacity.

- $O(1)$
- $O(n)$
- $O(n \log n)$
- $O(n^2)$ ✔

Submit You have used 1 of 1 attempt

Question 2

1/1 point (graded)

Let's imagine we add support to our dynamic array for a new operation PopBack (which removes the last element). Calling PopBack on an empty dynamic array is an error.

PopBack reallocates the dynamically-allocated array to a new array of half the capacity if the size is ≤ the capacity / 4 . So, for example, if, before a PopBack the size were 5 and the capacity were 8, then after the PopBack, the size would be 4 and the capacity would be 8. Only after two more PopBack when the size went down to 2 would the capacity go down to 4.

We want to consider the worst-case sequence of any n PushBack and PopBack operations, starting with an empty dynamic array.

What potential function results in the best possible amortized cost per operation?
Recall that the PushBack operation in a standard dynamic array increases its capacity twice when its size exceeds capacity.

- $\Phi(h) = 2 \times size - capacity$, it results in amortized cost $O(1)$
- $\Phi(h) = max(2 \times size - capacity, capacity/2 - size)$, it results in amortized cost $O(1)$ ✔
- $\Phi(h) = 2$, it results in amortized cost $O(1)$
- $\Phi(h) = 2$, it results in amortized cost $O(n)$

Submit You have used 1 of 1 attempt