

Practice Quiz 7

CSCI 136: Spring 2022

Your name: _____

Welcome back!

This week's quizzes cover readings, handouts, labs, and lecture materials up to and including April 4th. Answer the following questions as practice for your graded quiz on Friday—and as a warmup while coming back from break!

1. Let's say we have a `List` (recall that `List` is an interface with methods like `addLast`, `removeLast`, and `removeFirst`; it is implemented by `Vector`, `SinglyLinkedList`, and `DoublyLinkedList`).

We call `addLast()` on the `List` n times. Then, we call `removeLast()` on the `List` n times.

- (a) What is the worst case time to complete all of these operations on a `List` that is implemented using a `Vector`? Assume that `Vector` doubles in size each time the underlying array runs out of room.

Your answer: $O(n)$. Each `removeLast()` is $O(1)$; growing a vector to n takes $O(n)$ in total as shown in class.

- (b) What is the worst case time to complete all of these operations on a `List` that is implemented using a `SinglyLinkedList`? Assume that `SinglyLinkedList` does not have a tail pointer.

Your answer: $O(n^2)$. Each `addLast()` and `removeLast()` call takes $O(n)$ operations.

- (c) What is the worst case time to complete all of these operations on a `List` that is implemented using a `DoublyLinkedList`? Assume that `DoublyLinkedList` has a tail pointer.

Your answer: $O(n)$. Each operation is $O(1)$, for $O(n)$ in total.

2. Let's say we have another `List`. We call `addLast()` on the `List` n times. Then, we call `removeFirst()` on the `List` n times.

- (a) What is the worst case time to complete all of these operations on a `List` that is implemented using a `Vector`? Assume that `Vector` doubles in size each time the underlying array runs out of room.

Your answer: Each `removeFirst()` is $O(n)$ leading to $O(n^2)$ total; total for all `addLast()` is $O(n)$. Total: $O(n^2 + n) = O(n^2)$.

- (b) What is the worst case time to complete all of these operations on a `List` that is implemented using a `SinglyLinkedList`? Assume that `SinglyLinkedList` does not have a tail pointer.

Your answer: Each `removeFirst()` is $O(1)$; each `addLast()` is $O(n)$; giving $O(n^2)$ overall.

- (c) What is the worst case time to complete all of these operations on a `List` that is implemented using a `DoublyLinkedList`? Assume that `DoublyLinkedList` has a tail pointer.

Your answer: Each `addLast()` and `removeFirst()` is $O(1)$, giving $O(n)$ in total.