1) Efficiency - Multiple Choice [12 points]

Goal: Answer the following questions about time complexity.

Requirements:

- Assume an optimal implementation, as taught in the course.
- **Do not** use amortized analysis unless specified.
- Do not assume internal access to the data structure unless specified.
- Choose the tightest Big-O bound possible for the scenario.

A.) Adding to the middle index of an ArrayList.

○ O(1) ○ O(log n) ○ O(n) ○ O(n log n) ○ O(n^2)

B.) Adding to the back of a CircularSinglyLinkedList.

○ O(1) ○ O(log n) ○ O(n) ○ O(n log n) ○ O(n^2)

C.) Adding to the third-to-last spot in a DoublyLinkedList.

○ O(1) ○ O(log n) ○ O(n) ○ O(n log n) ○ O(n^2)

D.) **Average** case of adding an element that is larger than every other data in the BST.

○ O(1) ○ O(log n) ○ O(n) ○ O(n log n) ○ O(n^2)

E.) Removing the oldest element from a Stack using the pop operation.

○ O(1) ○ O(log n) ○ O(n) ○ O(n log n) ○ O(n^2)

F.) Enqueueing an element to a LinkedList-backed Queue.

○ O(1) ○ O(log n) ○ O(n) ○ O(n log n) ○ O(n^2)

G.) Searching for a value in a BST.

○ O(1) ○ O(log n) ○ O(n) ○ O(n log n) ○ O(n^2)

2) Scenarios - Multiple Choice [9 points]

Goal: Answer the following data structure scenario questions.

Requirements:

- Follow the implementations taught in lecture when answering questions.
- Select one answer for each question unless otherwise specified.

A.) Krispy Kreme needs your help tracking their supply of donuts. They are constantly making fresh, hot donuts to add to their stock, which then cool at the same rate. Customers may either take the hottest, freshest donut or choose the coldest, most aged donut instead. However, they may not take donuts that are in the middle. Which data structure best represents the donuts?

○ Queue ○ Deque ○ DoublyLinkedList ○ Heap

B.) Halloween is approaching, and you want a way to track the candy you've collected and model the line of trick-or-treaters at each house. You hate digging for candy in your bucket, so you will only ever take candy off the very top. Additionally, you don't like wasting time, so once you commit to waiting at a house, you will not leave the line until you have received candy. Which combination of data structures best represents your candy bucket and the lines of trick-or-treaters waiting at each house?

○ Stack and Queue ○ Stack and Deque ○ Heap and Queue ○ Heap and Deque

C.) What operations would a DoublyLinkedList with only a tail pointer perform faster than a SinglyLinkedList with both a head pointer and a tail pointer? Select all that apply.

□ addToFront() □ addToBack() □ addThirdFromBack() □ removeFromBack()

D.) The Hamby Corporation wants to create a record system for their massive amount of sales. Lookups are rare and only happen in the case of a discrepancy, but hundreds of items are added every minute and even one slow addition could cost thousands of dollars in sales. What structure should they use to optimize adding items fast?

○ ArrayList ○ SinglyLinkedList ○ Deque ○ BST

E.) You have various CPUs numbered 0, 1, 2, …, etc. You want to store and increment the number of operations *each* CPU completes. Because there are so many tasks, the counts go up rapidly. What data structure could store each CPU's count efficiently?

○ ArrayList ○ SinglyLinkedList ○ Queue ○ BST

F.) You are a city planner for MARTA and want to design software that tracks the movement of trains on the circular perimeter of Atlanta. You need to store each train relative to the trains adjacent to it. What data structure can naturally represent a loop of trains?

○ ArrayList ○ LinkedList ○ CircularSinglyLinkedList ○ Binary Tree

G.) Diego is a white-hat hacker that has made a virus to (ethically) infect the computers of 1332 TAs. To avoid suspicion, he programmed the virus to only infect **two** other systems before shutting the original computer down. He wants to make a graphical representation of the virus as it travels through the computers. Which data structure is suitable?

○ ArrayList ○ LinkedList ○ CircularSinglyLinkedList ○ Binary Tree

H.) You are building a delivery dispatch system for Taco Tuesday. Delivery requests are added in the order they are received, and must be delivered in that exact order to ensure fairness. What data structure would best support this?

○ Array ○ Stack ○ Queue ○ Deque

I.) You are designing an undo/redo system for a drawing app. Users can undo their last action or redo a previously undone action. Which data structure would be best for supporting both operations efficiently?

○ BST ○ Stack ○ Queue ○ Deque