CS2852 Lab 1

Name:

Date:

Instructor:

# Part 1

## What is the advantage of using the List interface and avoiding direct references to the ArrayList and LinkedList classes?

Your answer here

## Why do you think we created the ItemRequest class?

Your answer here

## Why do you think we created the LibraryInventory class? If we wanted to make the simulation more realistic, how could we change this class?

Your answer here

# Part 2

## Create a table of your benchmark results.

Your answer here

## The requestItem() method adds the object to the end or back of the list. Did you notice a significant difference in run time between the two types of lists?

Your answer here

## For the benchmark of the nextRequest(true) method in which all items are fulfillable, the method will always remove and return the first item from the front of the list. Did you notice a significant difference in performance between the two types of lists?

Your answer here

## For the benchmark of the nextRequest(false) method in which no items are fulfillable, the method will search through all items in the list, find no matches, and return null. Did you notice a significant difference in performance between the two types of lists? (Note that depending on how you implemented the method, your benchmark results may vary.)

Your answer here

## Thinking back to your startup, do you think it will be more common for item requests to be fulfillable or not? Based on that, which type of list would you choose and why?

Your answer here

# Part 3

## Paste screenshots of the two graphs created by your GUI program.

Your answer here

## Is the performance of removeFromFront() similar for the two lists?

Your answer here

## Is the performance of getMiddleBenchmark() similar for the two lists?

Your answer here

## How do the differences in performance of the removeFrontBenchmark() and getMiddleBenchmark() methods for the two list types explain some of the performance differences of the WaitingList.nextRequest() method?

Your answer here