College of Engineering
CS261

Programming Assignment 3

Linked List Variations

Problem 1: Deque and bag implementation

First, complete the linked list implementation of the deque (as in Worksheet 19) and bag ADTs (Worksheet 22).

Files needed:

- linkedList.c
- linkedList.h
- linkedListMain.c
- makefilelinkedList.txt(after you download, rename it to makefile)

Problem 2: Comparison

The file **linkedListMain.c** contains code which does the following:

1. Takes an integer argument (say n) from the command line and adds that many elements to the bag.
2. Prints the memory in KB used by the bag.
3. Calls **contains()** with each integer 0 to n in a loop.
4. Prints the time taken by these **contains()** calls in milliseconds.

Your job is to compare, for various values of n:

1. the memory usage for bag - implemented using a linked list (Problem 1) vs. implemented using a dynamic array. Plot the memory usage of the two programs for various values of n starting from n=1000 to n=200,000; doubling the number of elements on each run.
2. the running time for bag **contains()** - implemented using a linked list (Problem 1) vs. implemented using a dynamic array. Plot the running times of the two programs for various values of n starting from n=1000 to n=200,000; doubling the number of elements on each run.

Note that the dynamic array must have a capacity of 1000 to start with. The policy is to double the size when the array is full. We're providing the dynamic array implementation below.

Then answer the following questions:

- Which of the implementations uses more memory? Explain why.
- Which of the implementations is the fastest? Explain why.
- Would you expect anything to change if the loop performed **remove()** instead of **contains()**? If so, what?

Important:

- Please run all memory usage and timing tests on flip for consistency.
- The memory calculation code runs on flip only. Please follow the instructions in **linkedListMain.c** and **dynamicArrayMain.c** to comment out the memory code if you develop your programs in Visual Studio.

Files needed:

- dynamicArray.h
- dynamicArray.c
- dynamicArrayMain.c
- makefiledynamicArray.txt (after you download, rename it to makefile)

Problem 3: Implementation of the deque ADT using a circularly linked list

For this problem, you will implement the Deque ADT with a Circularly-Doubly-Linked List with a Sentinel. As you know, the sentinel is a special link, does not contain a value, and should not be removed. Using a sentinel makes some linked list operations easier and cleaner in implementation. This list is circular, meaning the end points back to the beginning, thus one sentinel suffices. The header file and the implementation file for this approach are **cirListDeque.h** and **cirListDeque.c**, respectively. Complete the functions in cirListDeque.c.

Files needed:

- cirListDeque.h
- cirListDeque.c
- makefilecirListDeque.txt (rename to makefile after downloading)
- listDequeTest.c

Grading

Problem 1 (36pts)

- init 2
- _addLink 4
- addBack 2
- addFront 2
- front 2
- back 2
- _removeLink 4
- removeFront 2
- removeBack 2
- empty 2
- print 4
- contains 4
- remove 4

Problem 2 (24pts)

- timing plot for **contains()** 10
- memory plot 10
- answers to the three questions 4

Problem 3 (50pts)

- init 4
- _createLink 4
- _addLinkAfter 4
- addBack 3
- addFront 3
- front 3
- back 3
- _removeLink 4
- removeFront 3
- removeBack 3
- _free 4
- isEmpty 2
- print 4
- reverse 6

What to submit:
You should submit three files: linkedList.c, cirListDeque.c, and PDF file for problem 2. The PDF file should contain the two plots and answers to the three questions. Do not make modification to the header files or we won't be able to compile your code!