CS 240

Data Structures and Algorithms

Spring 2014

1 Pre-Lab 03

1.1 Bag

As with the previous lab, you are being provided with a .h file. You are also required to generate your own test files and write your own driver code to be sure that your code does not produce any errors. You are also required to write a Makefile to ease compilation during the labs.

Just as with the previous lab, you will be given test code at the beginning of your lab session on Tuesday. It will really benefit you to test all of your code on your own very thoroughly before the lab. Otherwise, you may not be able to finish the assignment during the lab period.

For each task, before you look at the associated header file that you are to implement, you should consider what your method signature will look like. This will help you to solidify what each question is asking before you begin to answer.

For this pre-lab you will implement the Bag ADT that we discussed in class. We will be using a typedef in order to make this class slightly more generic. The provided typedef is `typedef int Element;`. This will, at compile time, replace instances of the word Element with int.

Data:

- A capacity for the Bag
- A value that represents how full the Bag is
- A list of all items in the Bag

Operations:

- Create a Bag
- Destroy a Bag
- Add to the Bag (at the end)
- Get a specific item from the Bag (using [])
- Remove an item from the Bag (at the end)
- Search for where a given item is located in the Bag
- Determine the capacity of the Bag
- Determine how many things are in the Bag
- Determine if the Bag is empty
- Empty the bag
- Print the Bag's contents

1.2 Notes

For the sake of consistency, you should double the capacity of the Bag each time you try to fit a new item in a full Bag. This means that as the Bag gets full and we need to put more things in it, the Bag grows to suit our needs. If you currently have no space in the Bag at all, then increase the space available by one.

Pay attention to the difference between return by value and return by reference (Why might we prefer one to another in this case?).

To print a Bag, simply print `# items:` , where # is replaced with the number of items. Following that, on the same line, all of the contents of the bag from first to last and separated by spaces should be printed.

You can test your program for memory leaks using the `valgrind` program and the `memcheck` tool. More information on `valgrind` can be found in the man pages.