Data Structures Digest

Due Sunday March 9, 2014

Problem Description

This week your programming assignment does not involve any programming! Instead, you will create a *digest* of the data structures we have encountered so far in CS 62. Merriam-Webster defines a *digest* as:

> digest (noun): a summation or condensation of a body of information

Your assignment is to summarize and condense the content we've covered in this course regarding data structures. You will create a pdf document that could be used, for example, by another college student who was studying Data Structures. Your digest will be graded according to three criteria:

1. Accuracy - The minimum requirement for this assignment is that any information provided in your digest be accurate!
2. Completeness - A list of the data structures that should be included in your digest is provided below.
3. Quality of writing - Please do not underestimate the benefit of producing a well-written digest! Strive for clear and easily understood writing.

This is a timely assignment since next week is the midterm exam. As additional incentive for you to create the best digest you can – accurate, complete, and well-written – you are allowed to use your own digest as a cheat-sheet for the midterm[1]. See the submission instructions at the bottom of this writeup.

Digest Details

Your digest should contain one section for each of the data structures listed below.

Each section should begin with a table that summarizes the Big-O complexity of the data structure. Each row in the table should be one particular implementation of the data structure. Each column in the table should correspond to the possible operations that can be performed on the data structure. Each entry in the table should list the *worst-case* Big-O complexity of performing the operation given the particular implementation. As an example, see the table below for `ArrayList`:

	`add`	`addLast`	`remove`	`get`	`set`
array	O(n)	O(n)*	O(n)	O(1)	O(1)

Table 1: Summary of Big-O complexity for `ArrayList`.

Notice that I used an asterisk for `addLast` in the table above since it is worth explaining that the average-case complexity using the doubling strategy is in fact O(1).

After the table, you should give a one sentence (minimum) justification for each entry in the table. You can organize this however you want (using a bullet point list, using paragraphs, etc.). Remember that a picture

[1] I briefly considered collecting the digests at the beginning of the exam and then randomly handing them back out so that everyone used someone else's digest as a cheat sheet. On further reflection, I realized this might create hard feelings between students.

is worth a thousand words – it may be helpful to include not just a written justification but also a picture illustrating a particular operation. This is the minimum requirement for each data structure.

You are welcome to include any additional content that you think would be helpful. However, any additional content should be on-topic! For example, you might include an example of adding an element to a heap (using an array implementation) since this nicely shows why `add` on a heap has complexity $O(\log_2(n))$. However, you cannot include off-topic problems that you want to have with you for the midterm. For example, you can't include an example of an induction problem for your own reference on the exam! *If you are not sure whether certain content you want to include is allowable, please ask the professor.*

A list of data structures

Note that your book uses the term `Vector` in place of `ArrayList`. Here is the list of topics that should be included in your digest:

- ArrayLists
  - Implemented using arrays
  - Operations: add, addLast, remove, get, set, contains
- Linked lists
  - Singly linked lists implemented using pointers (head pointer only)
  - Doubly linked lists implemented using pointers (head and tail)
  - Operations: addFirst, removeLast, contains
  - (Optional: circular singly linked lists)
  - (Optional: circular doubly linked lists)
- Stacks
  - ArrayList implementation
  - Singly linked list implementation
  - Operations: push, pop, peek
- Queues
  - ArrayList implementation
  - Doubly linked list implementation (Note: your textbook uses a Circular list to implement a queue whereas in class we discussed using a Doubly linked list. Feel free to choose whichever makes more sense to you.)
  - array implementation (Note: This is covered in your textbook. In this implementation, the queue has a fixed size)
  - Operations: enqueue, dequeue, peek (also called getFirst)
- Binary Trees
  - Recursive implementation (i.e. using pointers)
  - array implementation
  - Operations: the operations you might perform on a binary tree depend upon the application. Instead, please contrast the Big-O *space* requirements of both implementations
- Priority Queue

  - ArrayList implementation (Note: this was not discussed in class but is presented in your textbook as an alternate implementation to heaps.)
  - heap implementation
  - Operations: add, remove

- Binary Search tree
  - Recursive implementation of a binary tree
  - Operations: add, remove, contains

- Splay tree
  - Recursive implementation of a binary tree
  - Operations: add, remove, contains

Academic Honesty

You will find information about these data structures in your book as well as your lecture notes from class. You can also use online resources, e.g. Wikipedia, to aid your understanding. However, all written content should be your own words! Do not simply copy sentences from some other source. This is called plagiarism.

In addition, feel free to discuss this assignment with other people in the class. A good rule-of-thumb however is that when you sit down to write your digest you should not be looking at anything that has the answer on it. For example, you shouldn't have another student's digest in front of you. Or you shouldn't have the answer written on a white board for you to copy down. So, talk together, but then go away and write using your own words.

In summary, every sentence in your digest should be your own wonderful prose!

As a professor, I always want to believe that my students would never violate the academic honesty policy. Unfortunately, it does happen. Please, do not disappoint me and yourself. The purpose of this assignment is to help you *learn* at a deep level the material we've covered in class. Plagiarism goes completely against this purpose. Better to turn in a half-finished digest that represents your best effort then to copy from another student or another source and fail the course!

Grading

Your digest will be graded based on the following criteria:

criterion	points
Accuracy	7
Completeness	7
Qualify of writing	7
General correctness	4

What to hand in

Sunday night by midnight, please submit a pdf of your digest. You should name your pdf, "digest_LastNameFirstName". You can also print a copy of your digest and bring it with you to class on Monday to use as a resource for the midterm exam. At the end of the exam, you will hand in both your quiz and your digest.