CSE374 – Spring 2021 – Fifth graded assignment

This assignment is done individually. You can search for information online, use your textbook, ask the instructor or Teaching Assistant for help. Working with anyone else is a non-authorized collaboration, that is, an act of academic dishonesty. If you have any doubt or question about what qualifies as academic dishonesty, please contact the instructor to make an informed decision.

Objective: Practice heaps.
Due date: Thursday May 6th 11:59pm. Assignments do not receive a grade after the due date.
How to submit: A ZIP file sent through the course website (no emails accepted).
ZIP your whole Eclipse project, name it "*Firstname_Lastname*_Assignment3.zip"
Submissions in other file formats, or using other environments, will not be graded.
You are responsible for ensuring that your submission was submitted. A quick way to check is to download it back from the website and open it.
Grading (9 pts): Every question is worth 1 point. There are 9 questions in total hence 9 points.

I. Overview

It's relatively rare that you have to edit the code for a heap directly, but your knowledge of how that code works will help you understand the efficiency of any *algorithm that uses a heap*. For instance, if you do not know that a linked list must start with the head pointer and make its way to a desired spot, then you also wouldn't know the cost of a get. This assignment focuses on interview questions that would require using a heap. In Java, the heap is provided by java.util.PriorityQueue so please rely on that class instead of using your own custom heap.

For a brief refresher on heap operations, please see: https://www.interviewcake.com/concept/java/heap

II. Getting used to alternative tools

All code and explanation written in this section must be in a class named ***AlternativeTools***

We want to write a function that takes two arguments: E[] array and int n, where E is a generic type. This function must display (using a print) up to the first *n* unique elements of the array. Here are two examples:

- Array 1 5 2 7 2 1 9 1 2 1, and n = 2. The first two unique elements are 5 and 7.
- Array 9 4 5 1 5 2 9 9, and n = 3. We will only display 4 1 because there are only two unique elements.

1) Write a solution using a nested loop and named getNestedUnique(E[] array, int n). The time complexity should be $\Theta(n^2)$ and space complexity $\Theta(1)$ since you display as you go.
2) Write getSortedUnique, which starts by sorting and then goes through the array in a single loop. The time complexity of sorting is $\Theta(n \log n)$, then your loop is in $\Theta(n)$. Space is still $\Theta(1)$.
3) Write getMapUnique, which has two loops. The first loop uses a map to count how many times each number appears. The second loop uses the map to display the first n unique elements. As a comment above your code, write its time and space complexity.
4) Write getHeapUnique. Similarly to solution 3, it will operate in two loops and use a map. However, the map here should be storing a pair: number of times each character appeared and the index of its first appearance. Your second loop should be over *the map, not the array*. Use a heap in the second loop.

III. Explaining and illustrating the use of a heap

*Your work should be in a **PDF** file named **TextAndImageQuestions***

1) Read pages 324 to 327 on heapsort. Also see https://www.youtube.com/watch?v=2DmK_H7IdTo
In your own words, briefly explain how heap sort works.
Show the details of how you sort the following two arrays:
- [12 6 10 5 1 9}
- [1 14 7 8 3]

Yes, there are applets online that can do it step by step. So you're expected to get the point on this question. But if this question comes up in your final exam or next quiz, you need to know how to do it yourself. So the most important aspect is that you learn to do this.

2) Explain whether we can turn a min-heap into a Binary Search Tree (BST) in Θ(log n).

IV. A seasonal problem

*All code and explanation written in this section must be in a class named **DeathComesKnocking***

Consider N houses, each of whom contains N_i healthy students. Each day, an infected student visits one of these houses to party. When a house is visited by the student, all of the students in it are exposed to the virus and half of them get sick so they're taken to the hospital (i.e. they would not be in the house the next day). The problem is to find the maximum number of students that can be exposed to the virus within a given number of days.

The function has two arguments: getInfections(int[] houses, int days). Here are some examples:

- Over 5 days, with houses initially containing [8, 2, 6, 4, 10] students, we'd infect up to 33.
- Over 3 days, with houses initially containing [5, 6] students, we'd infect up to 14.

The heap must be involved in your solution.

V. Brief summaries

*Your work should be in a **PDF** file named **Summaries***

Watch the short recording (which is immediately below the link for this assignment). Then, answer each of the following in one brief paragraph:

1) How would you find the maximum in a min heap?
2) How would you measure the extent to which a tree can efficiently be put in an array?

The short recording will give you answers, as narrated by the instructor. We are interested in hearing your own words here. The Turn-It-In software for plagiarism check will also be used to ensure that we're reading your words, rather than the words of a random person on the internet.

Questions? Existential doubts? Feel free to contact our teaching assistant, Elinore at: eavensek@miamioh.edu

"This magnificent butterfly finds a little heap of dirt and sits still on it; but man will never on his heap of mud keep still." – Joseph Conrad