

CMPSC-132: Programming and Computation II
Fall 2019

Lab #9

Due Date: 11/08/2019, 11:59PM EST

Read the instructions carefully before starting the assignment. Make sure your code follows the stated guidelines to ensure full credit for your work.

Instructions:

- The work in this lab must be completed alone and must be your own.
- **Download the starter code file from the LAB9 Assignment on Canvas. Do not change the function names or given started code on your script.**
- A doctest is provided as an example of code functionality. Getting the same result as the doctest does not guarantee full credit. You are responsible for debugging and testing your code with enough data, you can share ideas and testing code during your recitation class.
- Each function must return the output (Do not use print in your final submission, otherwise your submissions will receive a -1 point deduction)
- **Do not include test code outside any function in the upload. Printing unwanted or ill-formatted data to output will cause the test cases to fail. Remove all your testing code before uploading your file (You can also remove the doctest). Do not include the input() function in your submission.**

Goal:

[10 pts] In our video lecture, we discussed the abstract data type Binary Heap. A binary heap is a complete binary tree which satisfies the heap ordering property and can be implemented with an array. In a binary heap, we will always have the element with the maximum (or minimum) element in the root node of the heap. Implement the class MinHeap with the following operations:

- *parent(index)* returns the value of the parent of heap[index]. *index* is between 1 and the size of the heap. If $index \leq 1$ or $index > \text{size of the heap}$, it returns None.
- *leftChild(index)* returns the value of the left child of heap[index], *index* is between 1 and the size of the heap and returns None if there is no value
- *rightChild(index)* returns the value of the right child of heap[index], *index* is between 1 and the size of the heap and returns None if there is no value

*** Note that as discussed in the video lecture, the index range of the array implementation of a heap is given from 1:n, NOT 0:n-1, so to return the correct value, you must subtract 1 from the video lecture formulas

- *len(heap_object)* returns the number of items in the heap. It needs no parameters and returns an integer.
- *insert(item)* adds the item into the heap satisfying the min heap property. It modifies self.heap and returns nothing. You can use the first three methods to assist the process.
- *deleteMin()* removes the min element of the heap (the root node). It needs no parameters and **returns the value removed from the heap**. The heap satisfies the min heap property after deletion. (If you think on a recursive approach, you can create another method to help this method restore the min heap property). You can use the first three methods to assist the process.

***NOTE: You can't use heap functions from the Python library, otherwise no credit will be given.
You are not allowed to modify the given constructor***

Deliverables:

- Submit your code in a file name LAB9.py to the Lab9 GradeScope assignment before the due date