Students:

This content is controlled by your instructor, and is not zyBooks content. Direct questions or concerns about this content to your instructor. If you have any technical issues with the zyLab submission system, use the **Trouble with lab** button at the bottom of the lab.

61.2 Lab 3: Heap Sort

Heap sort is one of three major sorting algorithms: merge sort, quick sort, and heap sort. Two, merge sort and heap sort, sort in O(n log n). Quick sort averages O(n log n) but in some cases takes $O(n^2)$.

A *heap* is an array (list) representation of a binary tree that implements a priority queue. The root of the heap, `heap[0]` is the highest priority (aka first) element in the queue. Furthermore, both children of the root (and so on, recursively to the leaves) are also priority queues.

Conceptually, heap sort takes an unordered list and creates a heap, an O(n log n) operation. It then removes each element, one by one, from the heap (from the root), while maintaining the heap (priority queue) property of the heap. The process of keeping the heap in order is typically called *heapify*ing, or *up-heap bubbling*, and has a cost O(log n). Doing it n times has cost O(n log n). Thus heap sort is O(n log n) as it has 2 stages, each of cost O(n log n).

This week's assignment is to implement heap sort. Write a function `heapsort(hlist)`, which takes a list of integers as its argument and returns a new list, which is the sorted version (smallest to largest integer) of the list.

A few basic rules:

- `heapsort(None)` returns `None`
- `heapsort([])` returns `[]`
- `hlist` may have duplicate entries, such as `[7, 4, 3, 1, 7].heapsort(hlist)` should include the duplicates (in this case `7` and the return value should be `[1,3,4,7,7]`
- `hlist` should not be changed after a call to `heapsort()`.

You **must not** use the Python `sorted()` routine or any other publicly available sorting or heap-building routine in your solution but you may wish to use them in debugging your code.

Python Tips

None for this assignment.

More on Heap Sort

Heap sort is discussed in the textbook in chapter 9 (which is part of your reading). I find the presentation in Cormen easier to understand and have arranged for that to be placed in the course reserves (see Module 0 in Canvas).

563960.3542450.qx3zqy7

LAB ACTIVITY | 61.2.1: Lab 3: Heap Sort | 36 / 45

Submission Instructions

Downloadable files

`main.py` , `tox.ini` , and `pycodestyle.py` **Download**

Upload your files below by dragging and dropping into the area or choosing a file on your hard drive.

main.py

✓ File added **Remove**

Submit for grading 4:09 minutes until next submission is allowed

Coding trail of your work What is this?

```
9/24 T8,8,36 min:4
```

Latest submission - 1:06 AM EDT on 09/24/24 Total score: 36 / 45

□ Only show failing tests **Download this submission**

1:Argument Test	*Details hidden by your instructor*	4 / 4
2:Heapifying Stress Test	*Details hidden by your instructor*	15 / 15
3:Scalability Test	*Details hidden by your instructor*	0 / 9
4:Sort List with Repeated Element	*Details hidden by your instructor*	9 / 9
5:PEP8 compliance		5 / 5

Your test produced no output.

6:Sanity Check 1 / 1

Test feedback

```
was able to load heapsort() and it returne
```

7:Immutability Test 2 / 2

Your output

```
Test passed!
```

Previous submissions

12:53 AM on 9/24/24	8 / 45	View
12:07 AM on 9/24/24	8 / 45	View

Trouble with lab?