

## Homework 5: Heaps

Due: Friday, February 28 at 5pm on Canvas

**Concepts:** sorting, heaps, hypothesis testing

- (15 points) Implement a **min-heap** that supports the following operations with the given runtime in parantheses:
  - Initialization** - given a (possibly empty) list of elements initialize a heap in-place ( $O(n)$ )
  - Length** - return the number of elements in the heap ( $O(1)$ )
  - Insert** - insert a new element into the heap ( $O(\log n)$ )
  - Find Min** - return the minimum value in the heap ( $O(1)$ )
  - Delete Min** - delete the minimum value element in the heap ( $O(\log n)$ )
  - Sorted List** - returns a new list containing all the elements from min to max ( $O(n \log n)$ )

More documentation is given in the file `hw5.py` on Canvas. Your implementation for initialization, insertion, deletion, and returning a sorted list will each count for three points. Additionally, there will be three points for clarity of your explanation at your check-off. You **do not** have to implement unit tests (see the problem below).

- (6 points) The **hypothesis** library in python is a helpful library to test your code because it generates instances to test your assertions on. Further, this library works with **pytest**. As an example, the code below generates a random list of integers that the test function for `length` is called on.

```
import pytest
from hypothesis import given
import hypothesis.strategies as st

@given(st.lists(st.integers()))
def test_heap_len(l):
    h = Heap(l)
    assert len(l) == h.length()
```

If we then run `pytest yourfile.py` the test function `test_heap_len` will be run for many generated lists `l`. For more documentation, see <https://hypothesis.readthedocs.io/en/latest/quickstart.html>.

You will use this package to test your implementation. At your check-off you should be able to answer: **how did you set up the tests?** what **purpose does each assertion** serve? are there **any weaknesses to your tests?**

- (2 points) Use the **hypothesis** package to **test sequentially inserting integers into a heap** and then **sequentially deleting the minimum until the heap is empty**.
- (2 points) Use the **hypothesis** package to **test initializing a heap with a list** and then **sequentially deleting the minimum until the heap is empty**.
- (2 points) Use the **hypothesis** package to **test initializing a heap with a list** and then **returning a sorted version of that list**.