Dr.Erdem KAYA
ekaya@metu.edu.tr
Mehmet Taha ŞAHİN
staha@metu.edu.tr

Homework 08
METU CENG310 Spring 2021
Data Structures and Algorithms with Python

Start Date: May 27th, 2021
Due Date: June 9th, 2021

1 Heap Performance Experiment

In this assignment, you will be conducting a simple experiment to show that removing minimum operations, in average, take time proportional to the binary logarithm of the input size. In doing so, you will

- Create a `HeapPriorityQueue` class by making use of Code Fragments 9.1, 9.4, and 9.5 from the textbook,
- Develop a function called `test_heap` that takes a single argument, `n`, which is the number of (key, value) pairs to be inserted to an instance of `HeapPriorityQueue` class, and then perform remove minimum operation on the heap object with each of the n items, and return the average and worst remove minimum times, and
- Develop a `report_results` function that creates and saves a line chart where x-axis is `n` and the y-axis is the average and worst remove minimum times.

1.1 The `test_heap` Function

The `test_heap` function should work as follows. Initially, it should create an instance of `HeapPriorityQueue` class (Let's call it `_heap`). And then `n` *distinct* random integer numbers should be drawn from the range $[0, 10^3 n]$, and inserted to `_heap` object with the `add` operation (you can add the same number as key and value). At this moment, your heap should have n *distinct* numbers.

After filling the heap, you should perform remove minimum (`remove_min`) operation until the heap is empty. You will need to measure the time elapsed between the beginning and the end of the `remove_min` operation. And those elapsed time values should be saved in a temporary list called `measures`. Elapsed times for each `remove_min` operations can take time in the magnitude of nanoseconds. So you may want to save the time value in nanoseconds unit rather than seconds for the sake of simplicity.

Finally, `test_heap` function should return the average and maximum of values in `measures` as a tuple.

1.2 The `report_results` Function

`report_results` is a function that does not take a parameter, however, reports the results of a series of tests that you will perform with `test_heap`. The `test_heap` function will be called iteratively for the range of `range(10000:1000001:10000)` (i.e., there will be 100 iterations). After each iteration, a tuple with the following values will be appended to a list object: `(n, avg_time, max_time)`. Let's call that list object `results_list`.

After having the results from this simple experiment, you will plot these results as a line chart by making use of the library module `matplotlib`. You are expected to learn how you can plot with `plot` function of `matplotlib` (`https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html`).

`plot` function accepts one-dimensional number arrays as parameters. You can convert the `results_list` to a `pandas` data frame and then work with value vectors as follows.

```
# import pandas to use pandas DataFrame
import pandas as pd
# create DataFrame using data
df = pd.DataFrame(results_list, columns =['n', 'avg', 'max'])

# Then you can use df['n'], df['avg'], and df['max'] as value vectors.
# matplotlib's plot function accepts value vectors, as you will find
    out.
```

The outcome of the execution of `report_results` function should be a single PDF file having a single line chart having two lines for average and maximum values plotted against `n`.

1.3 The Deliverable

Your deliverable should be a Python module named as `hw8.py` that includes `HeapPriorityQueue` class, and `test_heap` and `report_results` functions. You can also send your own PDF result for control purposes.

2 Instructions

A Homework-08 page will be generated soon after the start date of this homework. All deliveries should be done over ODTUClass. Please also be aware of the late penalties (Please check the Announcements – Homework and Assignment Policy in ODTUClass if you have not already done so.). Should you have any questions pertaining to homework tasks, please ask them in advance (not on the due date) for your own convenience.