Harrisburg University of Science & Technology

CISC 610 Data Structures & Algorithms

Assignment 4: Heaps

You have been provided a Python file, `heap.py`, which constructs a min-heap structure with a list. Using that code as a guide:

- Develop a `LinkedHeap` data structure in a file named `LinkedHeap` with the extension appropriate to your chosen langauge using a linked tree structure (Nodes and Pointers).
  - If using Python, you **MUST** use **and not alter** the `BinaryNode` provided in the lab.
  - If using another language, you **MUST** recode an equivalent `BinaryNode` object.
    - Regardless of language, the key is that the publicly visible interface is the same and that you must make it so that the element of the node is immutable, meaning that moving an element in the heap **MUST** be done by moving the entire node and changing all pointers.
- The heap must support the following interface:
  - `insert(key, value)`
    - This method/subroutine inserts the (key, value) pair into the appropriate postion within the min heap.
  - `delete()`
    - This method/subroutine deletes and returns a (key, value) pair with the minimum key value within the heap.
  - `peek()`
    - This method/subroutine returns a (key, value) pair with the minimum key value within the heap without altering the contents of the heap.
  - You may add any additional **private** helper methods that you see fit.
- All operations most abide by the rules that govern a heap (see lecture slides for reference).

Once you have your heap structure created, next you must use it as a backing structure to a priority queue.

- Develop a `PriorityQueue` data structure in a file named `PriorityQueue` that is backed by a linked heap.
- Implement the normal methods that accompany a priority queue structure:
  - `add(key, value)`
    - This method/subroutine adds the (key, value) pair into the appropriate postion within the min heap.
  - `remove()`
    - This method/subroutine removes and returns a (key, value) pair with the minimum key value within the heap.
  - `min()`
    - This method/subroutine returns a (key, value) pair with the minimum key within the heap but leaves the heap unaltered.
  - `is_empty()`
    - This method/subroutine returns True (or the langauge-specific equivalent) if the heap contains no key/value pairs, False (or the langauge-specific equivalent) otherwise.
  - `len(pq)` (or the langauge-specific equivalent)
    - This method/subroutine/function call returns the number of (key, value) pairs stored in the heap using the methodology appropriate to the language in which this structure is implemented.
- Perform the following operations to showcase your working priority queue:
  - Enqueue the following items: 11, 7, 8, 6, 5, 9, 4
  - Dequeue 3 items by priority, they should be 4, 5, & 6.

PLEASE NOTE: Your grade for this question will be based on your ability to meet the design specifications above. You have broad leeway with internal design decisions and the signatures of any helper subroutines you define. HOWEVER ANY SIGNATURE OR EXTERNAL BEHAVIOR THAT IS SPECIFIED ABOVE IS EXPECTED TO BE COMPLETED AS SPECIFIED. Failure to meet this specification will be penalized.

Your submission should be accompanied by a 8 minute walk-through of your code. This analysis should include your decision making process, the logic behind you code, an your original thoughts that went into the decision making on why your code is written and performs in the manner in which you have written it. If you can not adequately explain how your code functions, it is difficult to believe that you created it yourself as it is inherently difficult to make that which you don't understand.

All video submissions must:

- Be narrated by your own voice - Silent submissions will not be considered
  - If you need accommodations regarding your voice recording, reach out to me **BEFORE** the due date of the assignment
- Capture your screen to include the source code and other assets required by the assignment if necessary for the comprehension of your explanation.

Your submission should include:

- A link to your YouTube/Loom Video upload submitted as a `.txt` file or as a submission comment **OR** a video file (`.mp4` preferred).
  AND
- Your code project (source code, resource files, etc.) unzipped.
- If necessary, provide a README file if an explanation is required to execute your code.

Failure to submit your assignment as specified above will construed as non-submission and be given a 0. The instructor may, based solely on their discretion and judgement, deem the assignment to be completed in part and offer some credit for part or all of the assignment at a significant penalty. However, this is strictly under the purview of the instructor and should be assumed to be the exception, not the rule, in instances where the assignment is submitted incorrectly.