Mod 10 Lab: Priority Queues

Implement the Priority Queue ADT with two data structures:

- `PQ_UL` - unordered list
- `PQ_OL` - ordered list

Note: you can find partial or full solutions for this lab in the textbook and video lectures for this module - the problems were chosen because they are good learning exercises, and they double as good teaching exercises.

Avoid external resources *during* lab; do your best to figure things out with you and your partner. Feel free to access these resources afterwards if you do not finish during.

Priority Queue

Each priority queue you implement should support the following ADT

Magic Metohds

- `len`
  - returns the number of entries in the priority queue

Non-magic Methods

- `insert(item, priority)`
  - adds `item` with given `priority` to priority queue
- `find_min()`
  - returns (but does not remove) the `item` with minimum `priority`
- `remove_min()`
  - returns and removes the `item` with minimum `priority`

Special Cases

- Two entries in your priority queue should compare as equal if they have the same priority **and** the same item

Examples

Any examples below (next page) are intended to be illustrative, not exhaustive. Your code may have bugs even if it behaves as below. Write your own tests, and think carefully about edge cases.

Unordered list - sequential insertion

```
>>> from lab10 import *
>>> pq = PQ_UL()
>>> n = 100
>>> l = [i for i in range(n)]
>>> for i in range(n):
...     pq.insert(str(i), i)
...
>>> print(len(pq))
100
>>> old = pq.remove_min()
>>> for i in range(1, n):
...     peek = pq.find_min()
...     new = pq.remove_min()
...     assert new == peek
...     assert old.priority <= new.priority
...     old = new
...
>>> print(len(pq))
0
```

Ordered list - random insertion

```
>>> from lab10 import *
>>> import random
>>> pq = PQ_OL()
>>> n = 100
>>> l = [i for i in range(n)]
>>> random.shuffle(l)
>>> for i in range(n):
...     pq.insert(str(l[i]), l[i])
...
>>> print(len(pq))
100
>>> old = pq.remove_min()
>>> for i in range(1, n):
...     peek = pq.find_min()
...     new = pq.remove_min()
...     assert new == peek
...     assert old.priority <= new.priority
...     old = new
...
>>> print(len(pq))
0
```

Submitting

At a minimum, submit the following files:

- `lab10.py`
    - contains classes
        - `PQ_UL` - unordered list
        - `PQ_OL` - ordered list

Students must submit to Mimir **individually** by the due date (typically, two days after lab at 11:59 pm EST) to receive credit.

Grading

- 50 - `PQ_UL`
    - 25 - sequential insertion
    - 25 - random insertion
- 50- `PQ_OL`
    - 25 - sequential insertion
    - 25 - random insertion

Feedback

If you have any feedback on this assignment, please leave it here.

We check this feedback regularly. It has resulted in:

- A simplified, clear **Submitting** section on all assignments
- A simplified, clear **Grading** section on all assignments
- Clearer instructions on several assignments