reflection hw9

requirements for binary tree to be heap:

- heap order property: for min-heap, each node's value must be less than or equal to its children's values (for a max-heap, its the opposite). this makes sure the root is always the min or max element
- complete binary tree property: the tree has to be completely filled at all levels except the last level — which has to be filled from left to right. this makes sure the heap can be efficiently represented as an array and keeps the O(log n) time complexity for operations

heapq.heappush and ensures heap:

heapq.heappush adds new element to heap while maintaining heap properties

- appends new element to end of the array (keep complete tree property)
- bubble-up operation where new element is compared with parent and swapped if necessary to keep heap order property
- process continues up tree until heap property is satisfied

heapq.heappop and ensures heap:

heapq.heappop removes and returns the smallest element (root) while maintaining heap properties

- removes root element (min)
- moves last element in the array to root
- bubble-down operation where this element is compared with children and swapped with smaller child if necessary to maintain heap order property
- continues down tree until heap property is satisfied

PriorityQueue main output - min-heap:

```
len of pq = 0

Inserting items in order: 5, 3, 7, 1, 4
Current heap: [(1,'one'), (3,'three'), (7,'seven'), (5,'five'), (4,'four')]
Length: 5

Testing min and removeMin:
Current min: (1,one)
Removed min: (1,one)
Current heap: [(3,'three'), (4,'four'), (7,'seven'), (5,'five')]
New min: (3,three)

Testing empty queue behavior:
Caught EmptyError: Priority queue is empty
Caught EmptyError: Priority queue is empty
```

heap keeps the min-heap property even with insertion order

- heap aray is 1, 3, 7, 5, 4
- represent binary tree
  - root is 1
  - left child of 1 is 3
  - right child of 1 is 7
  - left child of 3 is 5
  - right child of 3 is 4

calendar.py output

```
--------------------
Calendar Menu:
1. Print Calendar
```

```
2. Insert New Event
3. Show Next Event
4. Remove Next Event
5. Exit
------------------
Enter your choice: 2
Enter event date/time (yyyy.mm.dd.hh:mm): 2025.06.04.17:15
Enter event description: meeting
------------------
Calendar Menu:
1. Print Calendar
2. Insert New Event
3. Show Next Event
4. Remove Next Event
5. Exit
------------------
Enter your choice: 2
Enter event date/time (yyyy.mm.dd.hh:mm): 2025.06.04.09:00
Enter event description: Morning
------------------
Calendar Menu:
1. Print Calendar
2. Insert New Event
3. Show Next Event
4. Remove Next Event
5. Exit
------------------
Enter your choice: 1
>>>>>>>>>>>>>>>>>>
Calendar Of Events:
2025/06/04 @ 09:00: Morning
2025/06/04 @ 17:15: meeting
<<<<<<<<<<<<<<<<<<
------------------
Calendar Menu:
1. Print Calendar
```

```
2. Insert New Event
3. Show Next Event
4. Remove Next Event
5. Exit
------------------
Enter your choice: 3
++++++++++++++++++++++++++++++++++++++
2025/06/04 @ 09:00: Morning
++++++++++++++++++++++++++++++++++++++
------------------
Calendar Menu:
1. Print Calendar
2. Insert New Event
3. Show Next Event
4. Remove Next Event
5. Exit
------------------
Enter your choice: 4
Removed from calendar: 2025/06/04 @ 09:00: Morning
------------------
Calendar Menu:
1. Print Calendar
2. Insert New Event
3. Show Next Event
4. Remove Next Event
5. Exit
------------------
Enter your choice: 1
>>>>>>>>>>>>>>>>>>
Calendar Of Events:
2025/06/04 @ 17:15: meeting
<<<<<<<<<<<<<<<<<<
```