CMPT 225

Lecture 27 – Priority Queue
Priority Queue

- Elements in a priority queue are queued in a sorted FIFO fashion using a priority value
- Priority value:
  - Numerical or other
  - Assigned to each element or can be attribute(s) of elements
- Dequeue: Removes the “highest” priority element
- Example:
Priority Queue Operations

- **isEmpty**: Is the priority queue empty?
- **getElementCount**: Returns the number of elements in the priority queue
- **enqueue**: Insert element into the priority queue
- **dequeue**: Remove element with highest priority from the priority queue
- **peek**: Retrieve element with highest priority from the priority queue (but does not remove the element)
- **dequeueAll**: Remove all elements from queue
Time efficiency analysis of Priority Queue’s operations

- Remember that a Queue’s operations enqueue() and dequeue() have a time efficiency of $O(1)$
- Priority Queue’s operations enqueue() and dequeue() cannot both have a time efficiency of $O(1)$ 😞
- For array-based and link-based implementations of PQ:
  - If enqueue is $O(1)$, then dequeue is $O(n)$
  - OR
  - If enqueue is $O(n)$, then dequeue is $O(1)$
- For heap-based implementation of PQ:
  - enqueue is $O(___________)$
  - dequeue is $O(___________)$
What about the FIFO characteristic of the Priority Queue

- FIFO can be seen as a Queue ADT class invariant
- Is it the case with a Priority Queue ADT class?
  - For array-based and link-based implementations of PQ?
  - For heap-based implementation of PQ?