Homework Assignment #1
Due: January 15, 2019 at 11:30 a.m.

1. A priority queue stores a bunch of items, each with a priority value. For simplicity, assume that items stored in the priority queue must be distinct. Let's consider a priority queue that provides an INSERT($item, priority$) operation, an EXTRACTMIN operation and an UPDATE($item, priority$) operation that changes the priority associated with $item$ to $priority$. Note that the latter operation is useful for some applications, like Dijkstra's algorithm.

(a) Should the formal description of a priority queue ADT be deterministic or non-deterministic? Why?

(b) Give a formal description of a priority queue ADT by specifying the set of states, etc. State any assumptions you are making about the priority queue when coming up with this formal description.

(c) Define a reasonable measure n of the size of the state of a priority queue.

(d) Briefly describe a simple data structure that implements a priority queue that achieves $\Theta(1)$ worst-case running time for EXTRACTMIN.

(e) One data structure for implementing a priority queue is a heap. Chapter 6 describes how to represent the heap as an array. However, the array representation requires us to know the maximum number of elements to be stored in the heap in advance (and the space used is proportional to this maximum size). An alternative way to represent a heap is a tree structure, using a node to represent each element of the heap, with pointers between nodes. What pointers do you need to store so that INSERT and EXTRACTMIN have worst-case time $O(\log n)$? You may assume that INSERTS satisfy the precondition that the inserted item is not already in the heap. Try to use as few pointers as possible. Draw a picture of the pointer structure for a heap containing elements $1, 7, 4, 3, 9, 6$.

Justify your answer by *briefly* describing how the INSERT and EXTRACTMIN would be implemented (you need not give very detailed algorithms).

(f) Let m be a natural number. Suppose that items to be stored in the priority queue come from the domain $D = \{1, 2, 3, \ldots, m\}$. Briefly describe a simple data structure that implements a priority queue that achieves $\Theta(1)$ worst-case running time for INSERT and UPDATE operations.

(g) What is the worst-case running time of EXTRACTMIN operations for the data structure described in part (f)? State your answer as a function of n and/or m using Θ notation.