COMPUTER SCIENCE 21A (FALL, 2018)
DATA STRUCTURES AND ALGORITHMS

PROBLEM SET 5

- Your assignment should be submitted via Latte as a PDF by the date it is due.
- Name your PDF file LastnameFirstname-PS5.pdf
- Use the provided template.
- Late submissions will not receive credit.

Q1.

Show how to implement the standard queue ADT using only a priority queue and one additional integer instance variable.

Q2.

A ***d*-ary** heap is like a binary heap, but instead of 2 children, nodes have *d* children.

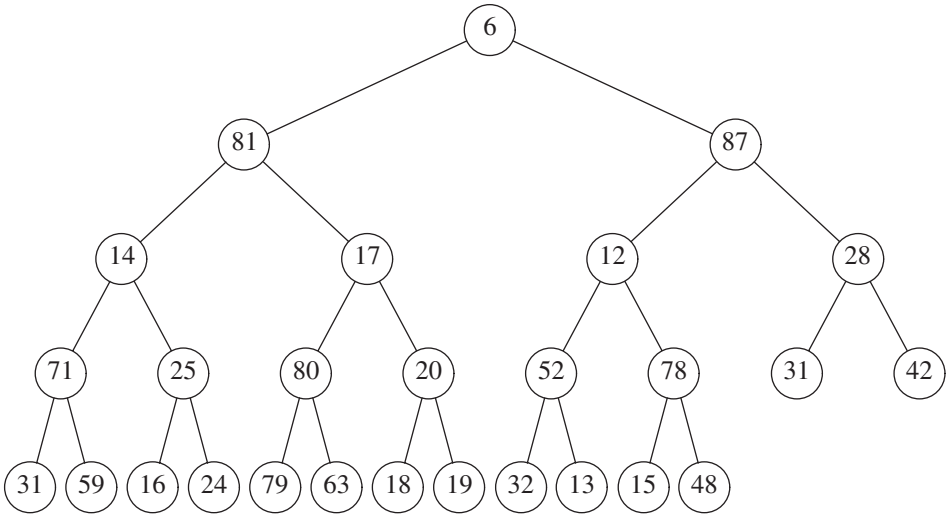
a. How would you represent a *d*-ary heap in an array?
b. What is the height of a *d*-ary heap of *n* elements in terms of *n* and *d*?
c. Give an efficient implementation of EXTRACT-MAX. Analyze its running time in terms of *d* and *n*.
d. Give an efficient implementation of HEAP-INCREASE-KEY(A, i, k), which sets A[*i*] ← max(A[*i*], *k*) and updates the heap structure appropriately. Analyze its running time in terms of *d* and *n*.
e.

Q3.

A group of children want to play a game, called Unmonopoly, where in each turn the player with the most money must give half of his/her money to the player with the least amount of money. What data structure(s) should be used to play this game efficiently? Why?

Q4.

A min-max heap is a data structure that supports both `deleteMin` and `deleteMax` in O(logN) per operation. The structure is identical to a binary heap, but the heap-order property is that for any node, X, at even depth, the element stored at X is smaller than the parent but larger than the grandparent, and for any node X at odd depth, the element stored at X is larger than the parent but smaller than the grandparent.



a. How do we find the minimum and maximum elements?
b. Give an algorithm to insert a new node into the min-max heap.
c. Give an algorithm to perform `deleteMin` and `deleteMax`.
d. Can you build a min-max heap in linear time?