CMSC351 (Kruskal) Homework 4 Due: Friday, March 2, 2018

Problem 1. Consider an array of size nine with the numbers in the following order

$$40, 20, 80, 60, 30, 10, 90, 50, 70.$$

(a) Phase 1: Form the heap. Show the heap as a tree. Show the heap as an array. Exactly how many comparisons did heap creation use? Make sure to form the heap bottom up as done in class.

(b) Phase 2: Start with the heap created in Part (a). Show the *array* after each element sifts down *after heap creation*. How many comparisons does each sift use? What is the total number of comparisons *excluding heap creation*?

Problem 2. A d-ary heap is like a binary heap, but instead of two children, nodes have d children. For each part *briefly justify* and/or *show your work* when appropriate.

(a) How would you represent a d-ary heap in an array?

(b) What is the index of leftmost child of the node stored at index i?

(c) What is the index of parent of the node stored at index i?

(d) What is the height of a d-ary heap of n elements in terms of n and d.

(e) Explain loosely (but clearly) how to extract the maximum element from the d-ary heap (and restore the heap). How many comparisons does it require?

(f) How many comparisons does it take to sort? Just get the high order term exactly, but show your calculations.

(g) What value(s) of d are optimal? Justify your answer.