CMPS 2200 Introduction to Algorithms – Fall 17

9/13/17

3. Homework

Due **9/20/17** at the beginning of class

Remember, you are allowed to turn in homeworks in groups of two. One writeup, with two names.

1. **Max-Heaps (7 points)**
   Justify your answers shortly.
   (a) (2 points) Where is the minimum element located in a max-heap? How can you compute it, and what is the runtime?
   (b) (2 points) Is an array that is sorted in decreasing order a max-heap? What about an array that is sorted in increasing order?
   (c) (3 points) List all valid binary max-heaps that store the numbers $1, 2, 3, 4$.

2. **d-Heaps (10 points)**
   A *d-ary max-heap*, d-heap for short, is the generalization of a binary heap to a d-ary tree, for a fixed $d \geq 2$. Every node can have up to d children, the tree has to be almost complete, and the max-heap property is fulfilled.
   (a) (2 points) For given fixed $h \geq 0$ and $d \geq 2$, give a formula for the number n of nodes in a complete d-ary tree of height h. Your formula should depend on n and d. Justify its correctness.
   (Hint: Use the geometric series.)
   (b) (2 points) Suppose a d-heap is stored in an array that begins with index 0. For an entry located at index i, in which location is its parent and in which locations are its children? (No formal proof necessary.)
   (c) (2 point) What is the height of a d-heap that contains n elements? The height should be a function of n and d. Shortly justify your answer; a formal proof is not necessary.
   (d) (2 points) Shortly explain how the insertion procedure works for d-heaps (you do not have to give pseudocode). What is the runtime of inserting an element into a d-heap of n elements? The runtime should be a function of n and d.
   (e) (2 points) Shortly explain how the extract_max procedure works for d-heaps (you do not have to give pseudocode). What is the runtime in terms of n and d, where n is the number of elements in the heap?