CMPS 6610 Algorithms – Fall 18

9/10/18

2. Homework

Due **9/17/18** at the beginning of class

1. **Heaps (9 points)**
   (a) (3 points) A (binary) min-heap is a binary tree that is almost complete and that fulfills the *min-heap* property: For every node x, we have $y \geq x$ for all nodes y in any subtree of x.

   Can you give an example of a binary tree that is both a max-heap and a min-heap? Find such an example for as many $n > 0$ as you can, where n is the number of elements in the binary tree. And argue for which n you cannot find such an example.

   (b) (3 points) Can you give an example of a binary tree that is both a max-heap and a binary search tree? Find such an example for as many $n > 0$ as you can, where n is the number of elements in the binary tree. And argue for which n you cannot find such an example.

   (c) (3 points) A sorting algorithm is stable if it preserves the input order among equal elements. Give a counterexample to show that heapsort is not stable.

2. **Divide-and-conquer (6 points)**
   Let $A = A[0, \ldots, n-1]$ be a sorted array of n distinct integers in the range 0 to n. The task is to identify the missing integer. For example, the missing integer in $A = [0, 1, 2, 3, 4, 6, 7]$ is 5. Give pseudocode for a divide-and-conquer algorithm that computes the missing integer in $O(\log n)$ time. Justify the correctness of your code and its runtime briefly.

3. **Recurrence (6 points)**
   Use a recursion tree and big-Oh induction to asymptotically solve the recurrence below:

$$\begin{aligned} T(0) &= T(1) = 1 \\ T(n) &= 2T(n/3) + T(n/4) + n, \quad \text{for all } n \geq 4 \end{aligned}$$

   You only need to prove the big-Oh bound. But make your bound as tight as possible. You do not need to prove the base case for the big-Oh induction.