Illinois Institute of Technology
Department of Computer Science

Homework Assignment 5

CS 430 Introduction to Algorithms
Spring Semester, 2017

Due: Monday, March 6

1. Exercise 17.3-3 on page 462 of CLRS3. (*Hint*: Consider a potential function proportional to the sum of the depths of the nodes in the heap.)

2. A *deque* (pronounced "deck") is like a queue, but one can insert and delete at either end—that is, one can insert/delete at both the head and the tail. The object of this problem is to implement a deque using three stacks, called *Head*, *Tail*, and *Temp*, in such a way that all insert/delete operations take amortized $O(1)$ time.

   (a) The two stacks *Head* and *Tail* contain, respectively, the front and rear elements of the deque. The tops of the stacks are the ends of the deque so they are accessible, while the bottom stack elements are the innermost of the deque. The four deque operations are insert/delete from the front and insert/delete from the rear. Describe the insert operations.

   (b) The delete operations are simple if the corresponding stack is not empty–just pop the top element off the appropriate stack. But if the stack you need to pop from is empty, you need to get to the bottom element on the other stack. Explain how to do this using the *Temp* stack to split the contents of the non-empty stack into two halves so that *Head* and *Tail* contain, respectively, the front and rear elements of the deque.

   (c) What is the worst-case cost of each of the four operations?

   (d) Using a potential function proportional to the absolute value of $|Head| - |Tail|$, show the *amortized time* of the four operations is $O(1)$. $|Head|$ and $|Tail|$ are, respectively, the numbers of elements on *Head* and *Tail*.