

Homework Assignment #4
Due: February 14, 2025 at 5:00 p.m.

Consider the Fibonacci heap data structure.

- [3] (a) Prove or disprove the following statement. The node with the maximum degree in a Fibonacci heap is always one of the roots. The first sentence of your answer should say whether you are proving or disproving the statement.
- [2] (b) Show that for all n there is a sequence of $O(n)$ INSERTS and DELETES starting with an empty Fibonacci heap that ends up with a heap containing a tree of height n .
- [6] (c) Describe how to implement a INCREASE-KEY(x, k) operation on a Fibonacci heap. Given a pointer to a node x , it increases the key of x to k . Remember that the operation may have to update the pointer to the minimum root of the Fibonacci heap. Your operation should have good amortized time and should not affect the amortized time of any of the other operations that we studied in class.

Hint: use an approach similar to DECREASEKEY for resolving violations of the heap-ordering property.

- [3] (d) **This part is for EECS5101 students only.** Is there a comparison-based implementation of an INCREASE-KEY operation on Fibonacci heaps that has amortized constant time per operation?