Homework Assignment #3
Due: October 7, 2022 at 11:59 p.m.

1. In a binomial heap, the children of a node are sorted according to the degrees of those children. Suppose that we wanted instead to order the children of each node according to the keys of the children. Also, the (singly-linked) list of roots will be sorted by key instead of by degree. (Assume all the keys in the binomial heap are distinct.)

To store the children of a node, we will use a *super-list* ADT (sorted by the children's keys) instead of a singly-linked list. A super-list stores a sorted sequence of elements and supports operations

- First returns first element in sequence.
- Successor(x) returns the element that comes after x in the sorted sequence (or null if x is the last element).
- Add(x) inserts x into the correct location in the sequence to maintain its sorted order.

We shall soon see how to implement a super-list so that the worst-case time for First and Successor is $O(1)$ and the worst-case time for Add on a sequence of length n is $O(\log n)$. For now, assume that you have a data structure that achieves these bounds.

(a) Describe a simple implementation of Union for the modified binomial heaps that takes $O(\log n \log\log n)$ time in the worst case, where n is the size of the larger heap being Unioned. You need not give a detailed proof of correctness, but you should explain why your algorithm is correct and why it achieves the stated running time.

(b) How long would the Insert and Extract-Min operations take in this modified version of binomial heaps?

(c) Describe how to implement Kth-smallest(H, k), which outputs the kth smallest key in the (modified) binomial heap H. The operation should not modify H. Your algorithm should run in $O(k \log k)$ time in the worst case. Explain why your algorithm is correct and why it achieves this running time.

2. Consider the Fibonacci heap data structure. 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.