CSC265 Fall 2020 Homework Assignment 7

due Tuesday, November 17, 2020

A **black&white heap** is a doubly-linked list of binomial trees that satisfies the following properties:

- The roots of the binomial trees in the list have strictly increasing degrees.
- Each node in the binomial trees is either white or black.
- The root of every binomial tree is white.
- If a white node is not the root of a binomial tree, its key is greater than or equal to the key of its parent.
- The black nodes all have different degrees.
- The degree of a black node is 1 less than the degree of its parent, i.e. the black node is the first child of its parent.

Each node in a black&white heap has pointers to its preceding sibling as well as its next sibling, provided they exist. If not, the pointers are NIL.

1. Suppose you have a linked-list H of binomial trees that satisfies all the properties of a black&white heap except that it has two black nodes b and b' of degree k both of which have parents of degree $k+1$. Given pointers to b and b', explain how to transform H in constant time so that it is a black&white heap with the same set of nodes except that, for some $k' > k$, it might have two black nodes of degree k' and it might have one black node of degree k' that has a parent of degree greater than $k'+1$.

2. Suppose you have a linked-list H of binomial trees that satisfies all the properties of a black&white heap except that it has exactly one black node b of degree k and its parent has degree greater than $k+1$. Suppose that b's sibling of degree $k+1$ is white. Given a pointer to b, explain how to transform H in constant time into black&white heap with the same set of nodes.

3. Suppose you have a linked-list H of binomial trees that satisfies all the properties of a black&white heap except that it has exactly one black node b of degree k and its parent has degree greater than $k+1$. Suppose that b's sibling of degree $k+1$ of b is black. Given a pointer to b, show how to transform H in constant time so that it is a black&white heap with the same set of nodes except that, for some $k' > k$, it might have two black nodes of degree k' and it might have one black node of degree k' that has a parent of degree greater than $k'+1$.

4. Suppose you have a linked-list H of binomial trees that satisfies all the properties of a black&white heap except that it has one black node b of degree k whose parent p has degree greater than $k+1$ and another black node b' of degree k whose parent has degree $k+1$. Given pointers to b and b', explain how to transform H in constant time so that it is a black&white

heap with the same set of nodes except that, for some $k' > k$, it might have two black nodes of degree k' and it might have one black node of degree k' that has a parent of degree greater than $k' + 1$.

5. The operation $\textsc{Decrease-Key}(H, x, v)$ takes a node x in a black&white heap H and a value v and decreases the priority of x by v. Explain how to perform $\textsc{Decrease-Key}$ in a black&white heap so that its amortized cost is $O(1)$ with respect to a potential function that is some constant times the number of black nodes in the black&white heap.