Homework #3

(Due: May 1)

Task 1. [200 Points] Perfect Binary Heaps

A *perfect binary tree* (PBT) is a binary tree in which each internal node has two children and all leaves are at the last level. A PBT of height k (≥ 0) has exactly $2^{k+1} - 1$ nodes of which exactly 2^k are leaves. We denote a PBT of height k as T_{k+1}.

For $h \geq 1$, we define an *extended perfect binary tree* $\widehat{T}_k$ as a PBT T_k with one extra node as the parent of its root. We define $\widehat{T}_0$ to be a single node. Hence, $\widehat{T}_k$ has exactly 2^k nodes. Figure 1 shows $\widehat{T}_k$ for $0 \leq k \leq 4$.

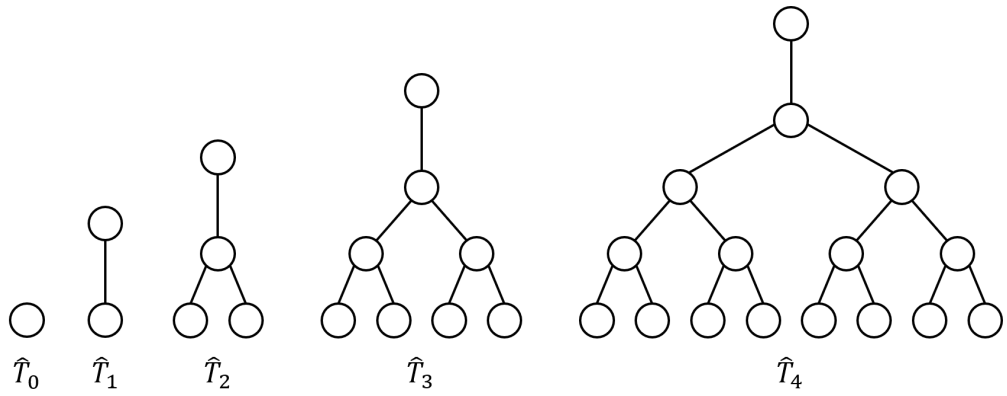

Figure 1: Extended perfect binary trees $\widehat{T}_k$ for $0 \leq k \leq 4$.

A *perfect binary heap* has exactly the same structure as a binomial heap, but each binomial tree B_k is replaced with a $\widehat{T}_k$ for all $k \geq 0$. The nodes in each $\widehat{T}_k$ satisfy the min-heap property.

(a) [**20 Points**] How do you efficiently merge two min-heap-ordered $\widehat{T}_k$'s to produce a min-heap-ordered $\widehat{T}_{k+1}$? What is the complexity of this merge operation?

(b) [**50 Points**] Analyze the amortized costs of MAKE-HEAP, MINIMUM, INSERT, EXTRACT-MIN, and UNION under eager union (i.e., using the array of pointers version of the data structure). Assuming that the data structure will never contain more than N items during its entire lifetime, express all complexities as functions of N.

(c) [**50 Points**] Repeat part (b) under lazy union (i.e., using the doubly linked list version of the data structure).

(*d*) [**40 Points**] Explain with justification how you will maintain and/or modify the data structure in part (*b*) so that all complexities can be expressed as functions of n, where n is the number of items currently in the data structure. Suppose you do not need to support UNION.

(*e*) [**40 Points**] Explain with justification how you will maintain and/or modify the data structure in part (*d*) so that INSERT can be supported in amortized $\mathcal{O}(1)$ time without increasing the asymptotic costs of other operations.