

Collaboration Allowed.

1. Consider the standard version of binomial queues as described in CLRS Chapter 19. For completeness I include here a sketch of this data structure in the half-ordered representation: a heap consists of a list of perfect half-ordered half trees, at most one per rank, in increasing order by rank, and a pointer to the minimum node. To insert an item, make it into a one-node tree and do fair matches with the existing trees until there is at most one tree per rank. (This process is like binary addition of 1.) To meld two heaps, do fair matches of their half trees by processing the half trees in increasing order by rank, until there is at most one half tree per rank. (This process is like binary addition of two numbers.) To do a minimum deletion, delete the minimum node. Disassemble the half tree previously rooted at this node, forming a new half tree rooted at each node along the path from the old ordered child of the minimum node descending through unordered children. (These half trees are in decreasing order by rank.) Do fair matches on these trees and the remaining half trees until there is at most one half tree per rank. We scale the time so that the time to do a fair match is at most 1. The worst-case time of an insertion is $O(1)$ plus the number of fair matches done, which is $O(\log n)$. The worst-case time of a minimum deletion is $O(\log n)$ plus the number of fair matches done, which is $O(\log n)$.

(a) Describe an implementation of meld whose worst-case time is $O(\min\{\log n_1, \log n_2\})$ plus the number of fair matches done, where n_1 and n_2 are the sizes of the two heaps being melded, and you can assume that n_1 and n_2 are at least two.

(b) Given your implementation in part (a), obtain the following amortized time bounds for the heap operations: $O(1)$ for insertion and melding, $O(\log n)$ for minimum deletion.

2. For either type-2 rank-pairing heaps or Fibonacci heaps, show that a key decrease operation on a heap of n nodes can take $\Omega(n)$ time. (For Fibonacci heaps, this is problem 20.4-1 in CLRS.)