
Efficient Algorithms and Datastructures I

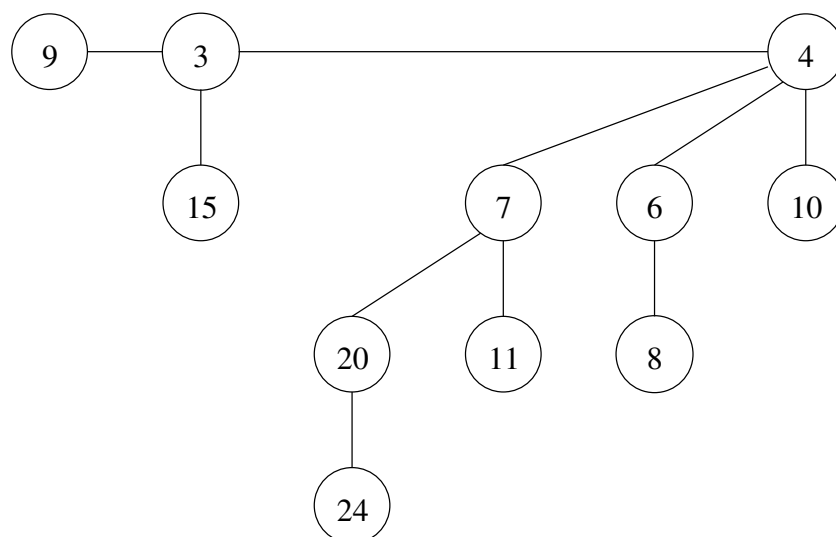
Question 1 (4 Points)

Consider a binary heap H implemented with a binary tree data structure (as implemented in the lectures) containing n items. Design an algorithm to find the k -th smallest item in H in $O(k \log k)$ time.

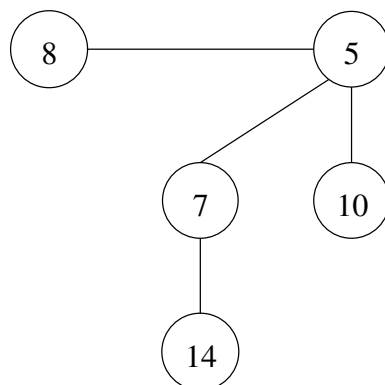
Question 2 (10 Points)

Consider the following Binomial Heaps:

Heap A:



Heap B:



Carry out the following operations sequentially on the heaps and show them after each operation(always carry out each operation on the result of the previous operation):

1. merge(A,B)
2. deleteMin()

Question 3 (10 Points)

We say that $f(n) = \tilde{\Omega}(g(n))$ if there exists a positive constant c such that $f(n) \geq cg(n) \geq 0$ for infinitely many integers n . Find inputs that cause DELETE-MIN, DECREASE-KEY, and DELETE to run in $\Omega(\log n)$ time for a binomial heap. Explain why the worst-case running times of INSERT, MINIMUM, and MERGE are $\tilde{\Omega}(\log n)$ but not $\Omega(\log n)$ for a binomial heap.