Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Harald Räcke
Chintan Shah

Wintersemester 2014-15
Assignment 8
December 1, 2014

Efficient Algorithms and Datastructures I

Question 1 (10 Points)

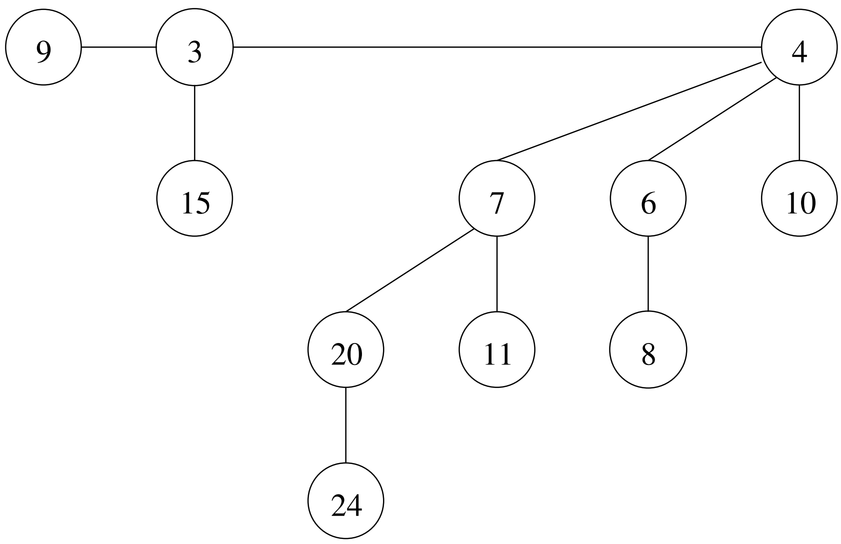
In hashing with chaining, if we modify the chaining scheme so that each list is kept in sorted order, how does it affect the running time for successful searches, unsuccessful searches, insertions, and deletions?

Question 2 (10 Points)

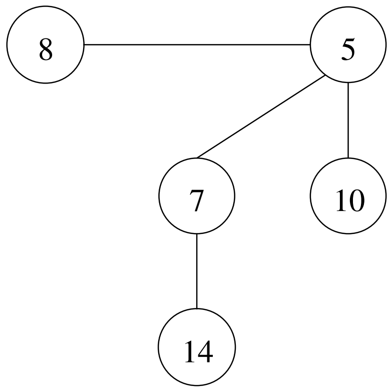
A forest is an undirected cycle-free graph, i.e. a forest is a graph, all of whose connected components are trees. A random graph is obtained by starting with a set of n vertices and adding edges between them at random. In the $G(n, p)$ model, for every pair of vertices $\{a, b\}$, the edge (a, b) occurs independently with probability p. Your goal is to show that for suitable value of p, the probability that G is not a forest is at most a constant (say $\leq \frac{1}{2}$). Note that G not being a forest means that $\exists$ a set $S \subseteq V, |S| = k$, such that the graph induced by S contains at least k edges. Of course, this trivially holds if for example you choose $p = 0$. You should aim for $p = \Omega\left(\frac{1}{n}\right)$. For example, $p = \frac{1}{4e^2 n}$ might be a good choice.

Question 3 (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 4 (10 Points)

We say that $f(n) = \overset{\infty}{\Omega}(g(n))$ 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 $\overset{\infty}{\Omega}(\log n)$ but not $\Omega(\log n)$ for a binomial heap.