Problems

1. A d-ary tree is a rooted tree in which each node has at most d children. Show that any d-ary tree with n nodes must have a depth of $\Omega(log\ n/log\ d)$. Can you give a precise formula for the minimum depth it could possibly have?

2. Sort each group of functions in increasing order of asymptotic (Big-O) growth.

   (a) $2^{2^{10}}n$, $1/n$, $log\ n$, n^2, 2^{10n}, $n(n-1)/2$, $n\sqrt{n}$, $n\ log\ n$

   (b) $n^{\sqrt{n}}$, 2^n, n^{10}, 3^n, $n^{log\ n}$, $\binom{n}{n/2}$, $log(n^n)$, $n!$

3. (a) Find (with proof) a function f_1 mapping positive integers to positive integers such that $f_1(2n)$ is $\mathcal{O}(f_1(n))$.

   (b) Find (with proof) a function f_2 mapping positive integers to positive integers such that $f_2(2n)$ is not $\mathcal{O}(f_2(n))$.

   (c) Prove that if $f(n)$ is $\mathcal{O}(g(n))$, and $g(n)$ is $\mathcal{O}(h(n))$, then $f(n)$ is $\mathcal{O}(h(n))$.

   (d) Prove or disprove: if f is not $\mathcal{O}(g)$, then g is $\mathcal{O}(f)$.

4. A binary tree is a rooted tree in which each node has at most two children. Show that in any binary tree the number of nodes with two children is exactly one less than the number of leaves. (Hint: induction.)

5. In this problem the input includes an array A such that $A[0 \dots n-1]$ contains n integers that are sorted into non-decreasing order: $A[i] \leq A[i+1]$ for $i = 0, 1, \dots, n-2$. The array A may contain repeated elements, e.g. $A = [0, 1, 1, 2, 3, 3, 3, 3, 4, 5, 5, 6, 6, 6]$

   (a) Describe carefully an algorithm $COUNT(A, x)$ that, given an array A, and an integer x, returns the number of occurrences of x in the array A. Your algorithm should be similar to binary search, and must run in $\mathcal{O}(log\ n)$.

   (b) Prove that your algorithm always terminates.

   (c) Prove that when your algorithm terminates, it terminates with the correct answer.

6. You are given a vertex-weighted graph. Consider the following definitions.

   Independent set: a set of vertices in a graph, no two of which are adjacent.

   Weight of independent set: sum of weights of vertices in the set.

   Max-weight independent set problem: Find an independent set which has maximum weight.

   Consider the following "greedy" approach to finding a max-weight independent set in a given vertex-weighted graph:

   1) Start with an empty set X.
   2) For each vertex v of the graph, in decreasing order of weight:

      add vertex v to the set X if v is not adjacent to any vertex in X.
   3) Return X, $weight(X)$.

   (a) Show that the "greedy" approach for the max-weight independent set is not optimal, by exhibiting a small counterexample.

   (b) How badly suboptimal can greed be, relative to optimal?