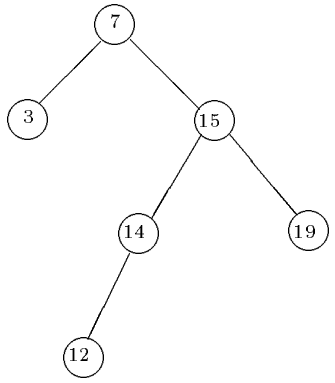
1. (6 marks total) Consider the binary search tree below. Using this tree, we will search for numbers chosen randomly from the set $\{1, 2, 3, ..., 20\}$. The numbers are not distributed uniformly. Instead, a number has probability 1/3 of being even and probability 2/3 of being odd, where all even numbers are equally likely and all odd numbers are equally likely.

   For the questions below, show your work and give numeric answers (i.e. no O notation).



   (a) (2 marks) What is the probability that a search will succeed?

   (b) (1 mark) What is the probability that a search will fail?

   (c) (3 marks) When performing a search, what is the expected number of nodes visited in this tree?

2. (10 marks total) A sequence of INSERT, DELETE and SEARCH operations is performed on a red-black tree whose size (number of nodes) never exceeds n. In addition, for the purposes of backup in case the computer crashes, after every n INSERT or DELETE operations, we perform a SAVETREE operation, which saves the present red-black tree in a file. The amount of time needed for a SAVETREE operation is proportional to the size of the tree.

   (a) (4 marks) What is the worst-case time complexity of a single operation in a sequence of m INSERT, DELETE, SEARCH and SAVETREE operations? Use Θ notation and justify your answer. You may assume that $m > n$.

   (b) (6 marks) For sequences of m INSERT, DELETE, SEARCH and SAVETREE operations, prove that the amortized time complexity of each SAVETREE operation is $O(1)$ and that the amortized time complexity of the other operations is $O(\log n)$. You may assume that $m > n$.

3. (8 marks) A *prioritized dictionary* is an abstract data type in which each object is a set of ordered triples, $\langle k, p, d \rangle$, where k is a key, p is a priority, and d is a data record. Both k and p are integers. In a prioritized dictionary, the same key may appear many times, but with different priorities. The operations on such a dictionary are

   - INSERT(k, p, d), which inserts the triple $\langle k, p, d \rangle$ into the dictionary;

CONTINUED

- DELETE(k), which deletes a triple with key k and maximal priority from the dictionary;
- SEARCH(k), which retrieves a triple with key k and maximal priority from the dictionary.

Describe how to implement a prioritized dictionary so that each of the three operations can be done in $O(\log n)$ time, where n is the number of triples in the dictionary. Justify your answer.

4. (10 marks) A large biological laboratory has a project to determine the complete DNA sequence of Human chromosome 7. The project will take several years to complete and will generate huge amounts of experimental data. This data is generated by chopping up very long strands of Human DNA into millions of shorter strands (called *clones*) and by performing experiments to determine which pairs of clones overlap. A very important part of the process is analyzing the data as it is generated to discover sets of clones that cover contiguous regions of the chromosome. Such sets are called *contigs*. The set of contigs is not fixed, but changes as more and more data is generated. Eventually, when enough clones are generated to cover the entire chromosome, and when enough overlaps are detected, there will be just one contig.

   The laboratory assigns a unique id number to each clone and it generates a stream of data items of the following two forms:

   - NEW I, which indicates a new clone has been discovered and given id I, and
   - OVERLAP I_1 I_2, which indicates that clones I_1 and I_2 overlap.

   Two clones, I and I', belong to the same contig if and only if there is a sequence of clones $I_1, I_2, ..., I_n$ such that clone I overlaps clone I_1, clone I' overlaps clone I_n, and clone I_{i-1} overlaps clone I_i for all $1 < i \leq n$.

   Each time a data item is generated, the set of contigs needs to be updated. Also, notification must be given whenever there is just one contig.
   Give an efficient algorithm for doing these two things. What is the amortized cost of your algorithm per data item? Justify your answer.

5. (8 marks) Suppose that A is an array of n positive integers, each with value less than m, where m is much larger than n (for example, $m \in \Omega(n^4)$). Give a randomized algorithm with worst-case expected running time $O(n)$ to find a number that occurs the most frequently in A. Briefly justify why your algorithm is correct and has the required running time.

6. (12 marks total) A binary tree is *ideally balanced* if, for every node x in the tree, the number of nodes in its two subtrees differs by at most 1.

   (a) (6 marks) Give an $O(n)$ time algorithm that, given a sorted array of n elements, constructs an ideally balanced binary search tree. Briefly justify the correctness of your algorithm and why it runs in $O(n)$ time in the worst case.

CONTINUED

(b) (3 marks) Give an $O(n)$ time algorithm that, given an n node binary search tree, transforms it into an ideally balanced binary search tree for the same elements. Briefly justify the correctness of your algorithm and why it runs in $O(n)$ time in the worst case.
You may use part (a) even if you didn't solve it.

(c) (3 marks) Prove that there is NO $O(n)$ time comparison based algorithm that, given an unsorted array of n elements, constructs an ideally balanced binary search tree.

7. (8 marks) We are given a directed graph $G = (V, E)$ that models a communication network. Each edge $(u, v) \in E$ has an associated real number $r(u, v)$ in the interval $[0, 1]$ that represents its *reliability*. In other words, $r(u, v)$ is the probability that the channel from u to v will not fail. We will assume that these probabilities are independent, so that the probability that none of the channels along a path fails is the product of the individual reliabilities.

   Give an efficient algorithm to find a most reliable path from one given vertex to another given vertex (*i.e.*, a path from the first vertex to the second that is least likely to fail). Briefly justify why your algorithm is correct and state what its running time is.

8. (5 marks) For general $n \geq 4$, describe a graph on n vertices and a weight function such that the graph has two edge-disjoint minimum spanning trees. Briefly explain why your construction is correct.
   Here, edge-disjoint means that there is no edge appearing in both of the minimum spanning trees.

9. BONUS (10 marks) The *median* of a set of n integers is the $\lceil n/2 \rceil$'th smallest element in the set. For example, 7 is the median of both $\{3, 7, 8, 9\}$ and $\{1, 3, 7, 8, 9\}$.

   Suppose you have an implementation of a priority queue that performs INSERT, MAXIMUM, and EXTRACT-MAX in worst case time $O(t(n))$, where n is the size of the priority queue and t is a non-decreasing function.

   Explain how this data structure can be used to construct a data structure for representing a set of size n so that MEDIAN can be performed in $O(1)$ time and INSERT can be performed in $O(t(n))$ time.

   Briefly justify the correctness of your algorithms and why they have the required running times.