

1. Describe the Binary Search Tree ordering property.
2. Starting with an empty Binary Search Tree, insert the values 35, 17, 88, 45, 51, 28 and 64 (in this order) into that tree. Sketch the resulting tree below.

3. What is the smallest possible height for a binary tree with 800 nodes?

What is the maximum height for a binary tree with 800 nodes?

4. Write a complete Java class definition for a class named `BTreeNode` describing a binary search tree node. Your class definition must define all the fields needed for objects of this type and a zero-parameter constructor that assigns default values to these fields. Make the fields public so that no additional accessor or mutator methods are needed. Use *Comparable* as the data type for the values stored in the nodes, since it will be necessary to compare these data values when the tree is accessed.



9. Write a complete Java class definition for a class named HashTableEntry describing an entry in a hash table. The description you write should be suitable for use in hash tables that support insert, find and delete operations.

10. Explain/describe the differences between *open-addressing* and *external-chaining* as hash table strategies.

11. Starting with an empty hash table with capacity = 19, and using division modulo 19 to compute hash values for integer keys, insert the following 14 items (in this order) into the hash table.

23, 62, 14, 73, 45, 29, 32, 16, 41, 74, 51, 39, 44, 66

Use a linear probe to handle collisions. Record your final result in the array below.

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]

12. What is the average look-up time for the values in the hash table you constructed above? Assume each value in the table has the same probability of being the target of a search.

13. What is a topological sort? What type of graph(s) can it be used on?

Construct an example of a topological sort, using a graph of your own choosing. Show one of the possible outputs that the topological sort could produce for your example.

14. Write part of the Java class definition for a class named Digraph describing a weighted directed graph. The description you write must define all necessary fields for this class. It must also define a one-parameter constructor whose parameter is an integer specifying the maximum number of vertices to be added to this graph. For our purposes in this question, a vertex will be represented by a Java String value.

In addition, you are to code an isEmpty() method that returns true if and only if there are no vertices in the graph and you are to code an indexOf(String v) method that returns the index of vertex v in the vertex list (or -1 if the vertex is not found).