1. Java's TreeSet implements a red-black tree. This guarantees that tree insertion and removal is no worse than $O(lgN)$ time. Compare this with HashSet implementation with $O(1)$ time for insertion and removal. Why, then, should we ever use TreeSet?

2. Show a BTree of order 4 that results from adding the following list of keys in order:
   
   (a) C G J X
   (b) C G J X N S U O A E B H I

3. Given a BTree of order 256, what is the maximum height of the tree if it contains 100,000 keys? Of order $M$?

4. Given the graph

   ![Graph](image)

   (a) Is the graph connected or unconnected? Explain.
   (b) What algorithm(s) can be used to determine if this graph is connected or unconnected?

5. In exactly what order would the vertices of this graph be visited using a breadth-first traversal beginning at node C? At node I? (assume neighbors are collected in alphabetical order.) Depth-first?