ADSA Quiz

Marks: 5

1. Which of the following is a true about Binary Trees
   a. Every binary tree is either complete or full
   b. Every complete binary tree is also a full binary tree
   c. Every full binary tree is also a complete binary tree
   d. No binary tree is both complete and full
   e. **None of the above**
2. What is the time complexity of Build Heap operation. Build Heap is used to build a max(or min) binary heap from a given array.
   a. O(nLogn)
   b. O(n^2)
   c. **O(n)**
   d. O(Logn)
3. What is common in three different types of traversals (Inorder, Preorder and Postorder)?
   a. Root is visited before right subtree
   b. **Left subtree is always visited before right subtree**
   c. Root is visited after left subtree
   d. All of the above
   e. None of the above
4. Which of the following statement(s) is TRUE?
   I. A hash function takes a message of arbitrary length and generates a fixed length code.
   II. A hash function takes a message of fixed length and generates a code of variable length.
   III. A hash function may give the same hash value for distinct messages.
   a. I only
   b. II and III only
   c. **I and III only**
   d. II only
5. How many undirected graphs (not necessarily connected) can be constructed out of a given set V= {V1, V2,…Vn} of n vertices ?
   a. n(n-l)/2
   b. 2^n
   c. n!
   d. **2^(n(n-1)/2)**
6. What is the maximum height of an AVL tree with p nodes?
   a. p
   b. **log(p)**
   c. log(p)/2
   d. p∕2
7. What are the operations that could be performed in O(log n) time complexity by red-black tree?
   a. **insertion, deletion, finding predecessor, successor**
   b. only insertion

     c. only finding predecessor, successor

     d. for sorting

8. Trie data structure is useful for
   a. Sorting numbers
   b. **Prefix matching**
   c. Number searching
   d. None of the above

9. Complete graph with n nodes will have the following number of edges
   a. n/2
   b. (n-1)/2
   c. n-1
   d. **n(n-1)/2**

10. The minimum spanning tree problem belongs to
    a. Divide and conquer
    b. **Greedy**
    c. Dynamic programming
    d. None of the above