

First Question (15)

State whether each of the following is *true* or *false*.

1. The correct choice of data structure allows major improvements in program Efficiency. ()
2. A data structure is the organization of data in a computer' s memory or in a disk file. ()
3. A binary search can be applied to an ordered array. ()
4. Linear searches don't require time proportional to the number of items in an array. ()
5. Sorting involves comparing the keys of data items in the array and moving the items (actually, references to the items) around until they' re in sorted order. ()
6. The bubble sort is the least efficient, but the simplest, sort. ()
7. A queue allows access to the first item that was inserted. ()
8. A stack allows access to the last item inserted. ()
9. A doubly linked list permits backward traversal and deletion from the end of the list. ()
10. A new link can be inserted before or after a link with a specified key value, following a traversal to find this link. ()
11. Some values of its arguments don't cause a recursive method to return without calling itself. This is called the base case. ()
12. A recursive approach may be inefficient. If so, it can sometimes be replaced with a simple loop or a stack-based approach. ()
13. A recursive method calls itself repeatedly, with different argument values each time. ()
14. Both triangular numbers and factorials can't be calculated using either a recursive method or a simple loop. ()
15. Edges are most commonly represented in a program by references to a node' s children (and sometimes to its parent). ()
16. Graphs can represent many real-world entities, including airline routes, electrical circuits, and job scheduling. ()

Second Question(15)

Choose the best Answer:

- 1) **Inserting an item into an unordered array**
 - a. takes time proportional to the size of the array.
 - b. requires multiple comparisons.
 - c. requires shifting other items to make room.
 - d. takes the same time no matter how many items there are.

- 2) **The maximum number of elements that must be examined to complete a binary search in an array of 200 elements is**
 - a. 200.
 - b. 8.
 - c. 1.
 - d. 13.

- 3) **A binary tree is a search tree if**
 - a. every non-leaf node has children whose key values are less than (or equal to) the parent.
 - b. every left child has a key less than the parent and every right child has a key greater than (or equal to) the parent.
 - c. in the path from the root to every leaf node, the key of each node is greater than (or equal to) the key of its parent.
 - d. a node can have a maximum of two children.

- 4) **A subtree of a binary tree always has**
 - a. a root that is a child of the main tree's root.
 - b. a root unconnected to the main tree's root.
 - c. fewer nodes than the main tree.
 - d. a sibling with the same number of nodes.

- 5) **Suppose you push 10, 20, 30, and 40 onto the stack. Then you pop three items. Which one is left on the stack?**
 - a. 40
 - b. 10
 - c. 20
 - d. 30

- 6) **Suppose you insert 15, 25, 35, and 45 into a queue. Then you remove three items. Which one is left?**
 - a. 45
 - b. 35
 - c. 15
 - d. 25

- 7) **Rearranging the contents of a data structure into a certain order is called**
 - a. Searching
 - b. Sorting.
 - c. Swapping.
 - d. None of the above.

- 8) **When you create a reference to a link in a linked list, it**
- a. must refer to the first link.
 - b. must refer to the link pointed to by current.
 - c. must refer to the link pointed to by next.
 - d. can refer to any link you want.

- 9) **The bubble sort algorithm alternates between**
- a. comparing and swapping.
 - b. moving and copying.
 - c. moving and comparing.
 - d. copying and comparing.

- 10) **Finding a node in a binary search tree involves going from node to node, asking**
- a. how big the node's key is in relation to the search key.
 - b. how big the node's key is compared to its right or left children.
 - c. what leaf node we want to reach.
 - d. what level we are on.

Third Question(20)

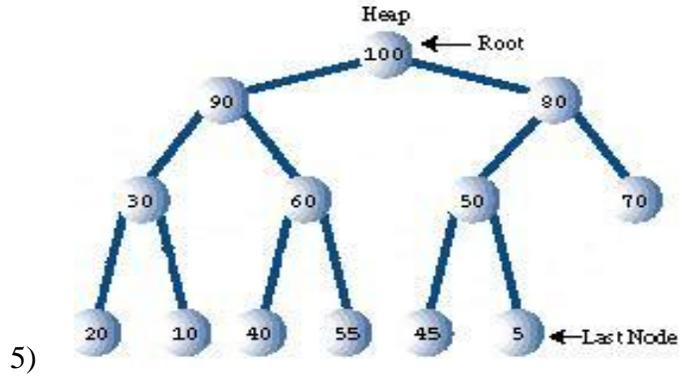
What is the Difference between each of the following terms, Justify with drawing if possible:

1) Depth First Search (DFS) Vs Breadth First Search (BFS)

2) Pop() Method Vs Top() Method

3) Singly Linked Lists Vs Double Linked Lists

4) fill in the array (Heap) as showing in the graph:

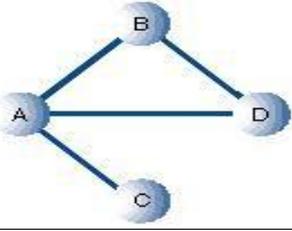


Fourth Question(5)

Sort the sequence 3, 1, 4, 1, 5, 9, 2, 6, 5 using insertion sort.

Fifth Question(5)

fill in the adjacency matrix as show in a graph

	A	B	C	D
A				
B				
C				
D				

sixth Question(10)

1) Removal PVD node form the Middle of a Doubly Linked List



2) Write Public method **insetElement** that insert element to order array using insertion sort.