## Section 1. **Fundamentals: Complexity, Algorthm Analysis**

1. An algorithm solves
   (a) A single problem or function
   (b) Multiple problems or functions
   (c) Has a single programming language implementation
2. A solution to a problem is said to be efficient if
   (a) the problem is solved within the required resource constraints.
   (b) problem is solved in constant (O(1)) time.
   (c) problem is solved regardless of the needed resources.
   (d) does not solve the problem.
3. A nested loop with each loop going from 1 through 1000 has a complexity of
   (a) O( 1 )
   (b) O( n)
   (c) O($n^3$)
   (d) None of the above
4. 3 nested loops with each loop going from 1 through N has a complexity of
   (a) O($n^2$)
   (b) O(1)
   (c) O($n^4$)
   (d) O($n^3$)
5. Which function grows slowest in terms of n?
   (a) $f(n) = n$
   (b) $f(n) = n^2$
   (c) $f(n) = log(n)$
   (d) $f(n) = 2^n$
6. Which function grows fastest, in terms of n?
   (a) f( n )= n log n
   (b) f( n )=$n^2$
   (c) f( n )=$n^{1000}$
   (d) f( n )=$2^n$
7. Which of the following function is $\Theta(n^2)$?
   (a) 10n
   (b) 5n+2
   (c) 10$n^2$+100n
   (d) 10$n^3$+100n

8. Which of the following statements are true?
   (a) n(n+1)/2 in O($n^3$)
   (b) n(n+1)/2 in O($n^2$)
   (c) n(n+1)/2 in O( n )
   (d) n(n+1)/2 in O( log n)
9. A program fragment that has exactly 1000 If statements has complexity of
   (a) O(1)
   (b) O(1000)
   (c) O( n )
   (d) O(1000 If)
10. Quadratic time is faster than
    (a) $\Theta(1)$
    (b) $\Theta(logn)$
    (c) $\Theta(n^2)$
    (d) $\Theta(n^3)$
11. Which function is $\Omega(n^3)$?
    (a) 10n
    (b) 5n+2
    (c) $10n^2 + 100n$
    (d) $10n^3 + 100n$
12. The code segment below has _____ time complexity?

```
for(int i = 0; i < n; i++){
  for(int j = 1; j < n: j=j*2){
     int val = (j*i);
     System.out.println(val)
   }
}
```

    (a) O(1)
    (b) O(n)
    (c) O($n^2$)
    (d) O(n log n)
13. If the function T(n) is said to be in O(g(n)), and c and n0 are positive constants, which statements are true?
    (a) T( n ) <= cg( n ) for all n >= n0
    (b) T( n ) >= cg( n ) for all n >= n0
    (c) T( n ) = cg( n ) for all n >= n0
    (d) none of above

14. If the function $T(n) \in \Omega(g(n)$, which statements are true?
    (a) T( n ) <= cg( n ) for all n >= n0, for positve c, n0
    (b) T( n ) >= cg( n ) for all n >= n0, for positve c, n0
    (c) T( n ) = cg( n ) for all n >= n0, for positve c, n0
    (d) None of above
15. If the function T(n) is said to be in $\Theta(g(n))$, which of the following statements are true?
    (a) T(n) is in O(n) and T(n) not in $\Omega(n)$
    (b) T(n) not in O( n ) and T(n) not in $\Omega(n)$
    (c) T(n) in O(n) and T(n) in $\Omega(n)$
    (d) T(n) = g(n)
16. A computational problem can be solved by
    (a) one algorithm
    (b) multiple algorithms
    (c) Java
    (d) C++
17. Big-O notation expresses
    (a) Upper bounds
    (b) Lower bounds
    (c) Average Case
    (d) Best case
18. Big-Omega notation expresses
    (a) Upper bounds
    (b) Lower bounds
    (c) Worst case
    (d) Best case
19. Big Theta notation expresses
    (a) Tight bounds
    (b) Lower bounds
    (c) Upper bounds
    (d) Best case
20. Any recursive definition must have a non-recursive part, called the _____ which permits the recursion to eventually terminate.
    (a) base case
    (b) foundation
    (c) primary case
    (d) first case

## Section 2. **Lists,Stacks,Queues**

1. On what principle does a stack work?
   (a) Last in first out(LIFO)
   (b) First in first out (FIFO)
   (c) Last in last out (LILO)
   (d) None of above
2. The terms "push" and "pop" generally refer to the
   (a) Array
   (b) Linked list
   (c) stack
   (d) All of above
3. (a) On what principle does a queue work?
   (b) Last in first out(LIFO)
   (c) First in first out (FIFO)
   (d) Last in never out(LINO)
   (e) None of above
4. All operations on stack are
   (a) O(1)
   (b) O(n)
   (c) O($n^2$)
   (d) None of above
5. The best case complexity for searching for a value in an array is
   (a) O( n )
   (b) O(log n)
   (c) O(1)
   (d) O(n!)
6. All operations on a queue are
   (a) O( 1 )
   (b) O( n )
   (c) O(log n)
   (d) None of above
7. Pushing an element into an linked list based stack results in the element inserted into
   (a) beginning of the linked list
   (b) end of the linked list
   (c) its correct position within the linked list.
   (d) None of the above

8. Pushing an element into an array based stack results in the element inserted into
   (a) beginning of the array
   (b) end of the array
   (c) its correct position within the linked list.
   (d) None of the above
9. Which of the following data structures can be used to reverse the order of set of data items?
   (a) queue
   (b) stack
   (c) software linked list
   (d) heap
10. Infix to Postfix expression conversion can be done using a
    (a) queue
    (b) stack
    (c) singly linked list
    (d) None of the above
11. Queue can be implemented by
    (a) Array
    (b) Linked list
    (c) Binary search tree
    (d) Hash table
12. The most natural data structure for maintaining the list of people wating in line at Starbucks is
    (a) Array
    (b) Stack
    (c) Queue
    (d) Linked List
13. The most natural data structure for handling recursive function calls is
    (a) Array
    (b) Stack
    (c) Queue
    (d) Linked List
14. An application requires use of a list where most of the operations are replacements of list objects at specific locations of the list. What type of list is appropriate?
    (a) Array
    (b) Singly Linked List
    (c) Doubly Linked List
    (d) Queue
    (e) Stack

15. Only _____ objects can be stored in an ordered list.
    (a) Ordered
    (b) Polymorphic
    (c) Comparable
16. A data structure that uses object reference variables to create links between objects is:
    (a) Linked Structure
    (b) Pointer
    (c) Self-referential
    (d) Array
17. A(n) _____ provides a means to iterate over a collection.
    (a) Search
    (b) Comparable
    (c) Iterator
    (d) toString
18. The worst case complexity for searching for a value in an array is
    (a) O(n)
    (b) O(log n)
    (c) O(1)
    (d) O(n!)
19. Replacing a value at the kth position of an array (A[k]) is
    (a) O( n )
    (b) O(log n)
    (c) $O(n^2)$
    (d) O(1)
20. Searching for a particular value in an array of n integers can be done in time
    (a) O(1)
    (b) O(n)
    (c) $O(n^2)$
21. Linked lists are
    (a) Linked structures
    (b) Linear structures
    (c) Dynamically allocated
    (d) All of the above

22. Retrieving the kth element of an array of n elements can be done in time
    (a) O(n)
    (b) O(1)
    (c) O(log n)
    (d) O(k)
23. Arrays are what type of structures?
    (a) Linear
    (b) Quadratic
    (c) Logarithmic
    (d) None of these
24. Deleting an element prior to the current element can be done in a singly linked list of n elements in time
    (a) O(1)
    (b) O(n)
    (c) O(log n)
    (d) $O(n^2)$
25. Declaring an array of 10 integers or a linked list that can contain upto 10 integers will take storage of (assuming 4 bytes for integers and 4 bytes for pointers)
    (a) 40 and 4 bytes
    (b) 4 and 40 bytes
    (c) 0 and 0 bytes
26. A node of a linked list usually contains two units of information. They are
    (a) Data only
    (b) Links only
    (c) Index
    (d) Data and Link
    (e) None of the above
27. An array of 10 integers and a singly linked list of 10 integers will consume (assume an integer or pointer takes up 4 bytes)
    (a) 40 and 40 bytes
    (b) 0 and 0 bytes
    (c) 40 and 80 bytes
    (d) 80 and 40 bytes
28. A node of a doubly linked list usually contains how many pointers?
    (a) 1
    (b) 2
    (c) 3
    (d) 4

29. An array of 10 integers and a doubly linked list of 10 integers will consume (assume an integer or pointer consumes 4 bytes each)
    (a) 0 and 0 bytes
    (b) 40 and 80 bytes
    (c) 80 and 40 bytes
    (d) 40 and 120 bytes
30. The last node in a singly linked list points to
    (a) Header
    (b) Null
    (c) Self
    (d) None of the above
31. The last node in a singly linked circular list points to
    (a) Header
    (b) First node
    (c) Last node
    (d) Self
32. An application requires the use of a list involving frequent insertions and deletions of elements anywhere in the list. What type of list is most appropriate?
    (a) Array
    (b) Singly linked list
    (c) Doubly linked list
    (d) Integer
33. Inserting an element into a queue is referred to as
    (a) Dequeue
    (b) Enqueue
    (c) Push
    (d) Pop
34. Removing an element from a queue is referred to as
    (a) Remove
    (b) Discard
    (c) Dequeue
    (d) Pop

## Section 3. **Binary Trees, Binary Search Trees, Heaps**

1. Each BinaryTreeNode object maintains a reference to the element stored at that node as well as references to each of the nodes _____
   (a) children
   (b) siblings
   (c) ancestors
   (d) parents
2. If a binary search tree is not _____ it may be less efficient than a linear structure.
   (a) complete
   (b) empty
   (c) balanced
   (d) None of the above
3. In general, a balanced n-ary tree with m elements will have height _____
   (a) $log_n m$
   (b) $log_m m$
   (c) $log_m n$
   (d) $log_n n$
4. Level order traversal of a binary tree can be implemented using a
   (a) queue
   (b) stack
   (c) linked list
   (d) binary search tree
5. Postorder traversal of a binary tree will visit
   (a) Node, then its children
   (b) Node's children, then the node
   (c) Node, left child, then right child
   (d) No particular order
6. Preorder traversal of a binary tree can be implemented using a
   (a) stack
   (b) queue
   (c) heap
   (d) singly linked list
7. Preorder traversal of a binary search tree with integer search keyswill visit nodes in
   (a) Ascending order
   (b) Descending order
   (c) No particular order
   (d) Ascending or Descending order

8. Most of Binary Search Tree algorithms(like insertion, search) typically run in time O( log n), if the tree is reasonably well balanced. What is n?
   (a) The depth of the tree.
   (b) The number of nodes in the tree.
   (c) The number of leaves.
   (d) The number of nodes at each level.
9. Which binary search tree traversal method will result in a sorted array of its search keys?
   (a) Preorder
   (b) Inorder
   (c) Postorder
   (d) Levelorder
10. What is the minimum number of nodes in a complete binary tree of height k? (root is at level 0).
    (a) $2^k - 1$
    (b) $2^{k-1} + 1$
    (c) $2^{k+1}$
    (d) $2^{k+1} - 1$
11. Which of the following statements hold true for binary search trees?
    (a) The left subtree of a node contains only nodes with keys less than the node's key
    (b) The right subtree of a node contains only nodes with keys greater than the node's key
    (c) The median of the key values is at the root of the tree.
    (d) All of above.
12. Huffman Coding trees code messages using
    (a) Variable length code
    (b) Constant length code
    (c) Balanced binary trees
    (d) None of the above.
13. Inorder traversal of a binary search tree containing integer search keys will visit nodes in
    (a) ascending order
    (b) descending order
    (c) no particular order
    (d) ascending or descending order
14. Deleting the leaf node a balanced binary search containing N nodes takes time
    (a) O(1)
    (b) O(N)
    (c) O($N^2$)
    (d) O(log N)

15. Inserting an element into a highly unbalanced binary search tree containing N nodes will take time
    (a) O(1)
    (b) O(N)
    (c) O($N^2$)
    (d) O(log N)
16. Preorder traversal of a binary search tree containing N nodes takes time
    (a) O(1)
    (b) O(N)
    (c) O($N^2$)
    (d) O(log N)
17. Deleting the root node of a balanced search tree takes
    (a) O(1)
    (b) O(n)
    (c) O(log n)
    (d) O($n^2$)
18. Finding the search key at the root node of a highly unbalanced binary search tree containing N nodes takes time
    (a) O(1)
    (b) O(N)
    (c) O($N^2$)
    (d) O(log N)
19. Which of following statements are true of a Max-heap?
    (a) It has no ordering.
    (b) All nodes have values greater than their child values.
    (c) All nodes have values less than child values.
    (d) The root stores the largest value.
20. Heaps are
    (a) Complete Trees
    (b) Full Trees
    (c) Binary Search Trees
    (d) All of the above.
21. A heap is represented using a
    (a) Binary search tree
    (b) Complete tree
    (c) Huffman Coding Tree
    (d) Min heap

22. Typically, in heap implementations, we keep track of the position of the last node or, more precisely, the last _____ in the tree.
    (a) root
    (b) internal node
    (c) leaf
    (d) tree

23. Since a heap is a _____ there is only one correct location for the insertion of a new node, and that is either the next open position from the left at level h or the first position on the left at level h+1 if level h is full.
    (a) Minheap
    (b) Maxheap
    (c) Balanced tree
    (d) Complete tree

24. In a Max-Heap, each node's priority value
    (a) Smaller or larger than all its children
    (b) Larger than all its children
    (c) Smaller than all its children
    (d) None of the above.

25. In a Min-Heap, each node's priority value
    (a) Smaller or larger than all its children
    (b) Larger than all its children
    (c) Smaller than all its children
    (d) None of the above.

26. A priority queues remove function always removes the node with the highest value. Which data structure can be used to implement it? Which data structure is used to implement it?
    (a) list
    (b) queue
    (c) stack
    (d) binary search tree
    (e) min heap
    (f) max heap

27. Inserting an element into a heap containing N nodes takes time
    (a) O(1)
    (b) O(N)
    (c) O(log N)
    (d) O($N^2$)

28. Deleting highest priority element from a max heap containing N nodes followed by reheaping takes time
    (a) O(1)
    (b) O(N)
    (c) O(log N)
    (d) O($N^2$)
29. Searching for a specific priority value in a Max Heap takes time
    (a) O(1)
    (b) O(N)
    (c) O(log N)
    (d) O($N^2$)
30. Finding the last node of a heap can done in
    (a) O(1)
    (b) O(N)
    (c) O(log N)
    (d) O($N^2$)

## Section 4. **Graphs**

1. An undirected graph is a graph where the pairings representing the edges are
    (a) Unordered
    (b) Ordered
    (c) Sorted
    (d) None of the above
2. Two vertices in a graph are _____if there is an edge connecting them.
    (a) Adjacent
    (b) Ordered
    (c) None of the above
    (d) Connected
3. 12. Two vertices in a graph are _____ if there is an edge connecting them.
    (a) Connected
    (b) Adjacent
    (c) Ordered
    (d) None of the above
4. A _____ is a sequence of edges that connects two vertices in a graph.
    (a) Connection
    (b) Path
    (c) Set
    (d) Cycle

5. An undirected tree is a connected, acyclic, undirected graph with one element designated as the _____
   (a) Starting point
   (b) Root
   (c) Leaf
   (d) Cycle
6. Graphs can be represented in 2 different ways. They are
   (a) Binary tree
   (b) Adjacency list
   (c) Adjacency matrix
   (d) Linked list
7. Depth-first search of a graph is best implemented using
   (a) a stack and recursion
   (b) a queue
   (c) a tree
   (d) a linked list
8. Breadth-first search of a graph is best implemented using
   (a) a tree
   (b) a stack
   (c) a queue
   (d) a linked list
9. Which representation is preferred for a sparsely connected graph?
   (a) Tree
   (b) Adjacency list
   (c) Adjacency matrix
   (d) Linked list
10. Which representation is preferred for a densely connected graph?
    (a) Tree
    (b) Adjacency list
    (c) Adjacency matrix
    (d) Linked list
11. Topological sort is a graph application that requires graphs to
    (a) be cyclic
    (b) Acyclic
    (c) a Minimum spanning tree

12. Traversing a connected component of a graph visits
    (a) All of the graph nodes
    (b) Some of the graph nodes
    (c) None of the graph nodes
    (d) No node

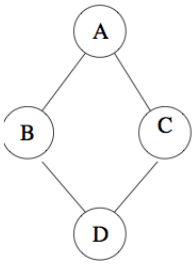
## Section 5. Hash Tables

1. Searching for an element in a a hash table containing n elements can be performed in time
    (a) O(1)
    (b) O( n )
    (c) O($n^2$)
    (d) None of above
2. Which of following methods can be used to solve the collision problem in hash tables?
    (a) Open hashing
    (b) Chaining
    (c) None of above
3. Searching for a range of values containing N elements in a hash table can be done in time
    (a) O(1)
    (b) O(N)
    (c) O(log N)
    (d) Cannot be done.
4. It is advisable to keep the hash table to be
    (a) Below 60 percent full.
    (b) Above 60 percent full.
    (c) 100 percent full
    (d) 90 percent full.

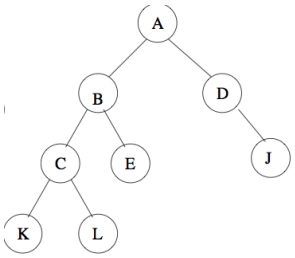
## Section 6. General: Comparing Data Structures

1. Select best data structure for this problem(assume data structure has been created prior to the operations to be performed): given a phone number return the name of the person.
    (a) Hash table
    (b) Linked list
    (c) Binary search tree
    (d) Heap

2. Select best data structure for this problem(assume data structure has been created prior to the operations to be performed): iterate through a list of the names in alphabetical order in a collection.
   (a) Hash table
   (b) Linked list
   (c) Binary search tree
   (d) Heap
3. Select best data structure for this problem (assume data structure has been created prior to the operations to be performed). Return the cell phone number that has had the most usage time.
   (a) Hash table
   (b) Linked list
   (c) Binary search tree
   (d) Max-heap
4. The structure in the following figure is

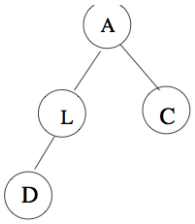



   (a) Binary Tree
   (b) Complete Tree
   (c) Heap
   (d) Not a Tree
5. The structure in the following figure is




   (a) full tree
   (b) Complete Tree
   (c) Binary Tree
   (d) Min Heap

6. The structure in the following figure is




(a) Binary Search Tree
(b) Full Tree
(c) Complete Tree
(d) Not a tree