

655: In a circular queue with rear and front pointers and CAPACITY as the queue array, how do you increment the rear end of the queue? B

- A.rear++
- B.(rear+1) % CAPACITY
- C.(rear % CAPACITY)+1
- D.None of the above is correct.

660: What is the time complexity of pop() operation when the stack is implemented using an array? A

- A.O(1)
- B.O(n)
- C.O(log n)
- D.O(n log n)

693: Consider the following operation performed on a stack of size 5: Push(1) Pop() Push(2) Push(3) Pop() Push(4) Pop() Pop() Push(5), After the completion of all operation, the number of elements present in stack is A

- A.1
- B.2
- C.3
- D.4

034: Problem abstraction is needed in which of the following situations? B

- A.To obtain and exact view of the problem at hand.
- B.To obtain an abstract view of the problem at hand.
- C.To merge necessary as well as unnecessary facts of real problems.
- D.To transform real problems into automated problems.

057: Which is the best way to go for searching in Game playing problems? C

- A.The linear search approach.
- B.The random search approach.
- C.The heuristic search approach.

D.None of the above is correct.

076: In the IT community, which of the following tasks require extensive knowledge of Data Structures: D

A.designing a new software component.

B.maintenance of a software component.

C.testing and improving existing software component.

D.all of the above are correct.

077: In real life, real problems are usually characterized by which of the following: C

A.they are usually automated by nature.

B.they are easy to analyze and handle.

C.they contain necessary as well as unnecessary details.

D.none of the above is correct.

078: Dynamic stacks can best be implemented with which of the following elementary data structures? A

A.Singly Linked Lists.

B.Doubly Linked Lists.

C.Single-Dimension arrays.

D.all of the above are correct.

079: The abstract view, or model, of real problems should include: C

A.data items that are affected.

B.operations identified by the problem.

C.both A and B.

D.none of the above is correct.

080: What best describe the operation “Constructor” of an Abstract Data Type (ADT)? D

A.It is called automatically whenever you create an object from the ADT.

B.One can define several constructors, all with the same name, but with different parameters.

C.There must be at least one constructor for a given ADT.

D.All of the above are correct.

081: An interface of an ADT is best defined to be: C

A.the set of operations provided by the ADT.

B.a subset of the operations defined by the ADT.

C.a subset of the operations defined by the ADT the user can access.

D.the set of private operations defined by the ADT the user can access.

082: An abstract data type is usually characterized by: D

A.it helps extending the language.

B.it defines a set of new operations.

C.it exports a new data type.

D.all of the above are correct.

083: Analyzing algorithms by running the programs representing them several times: B

A.will be informative in the general case.

B.will not be informative in the general case.

C.will give us its general behavior.

D.none of the above is correct.

084: Which of the following should be our main goal from analyzing algorithms? B

A.Predicting some future failures.

B.Estimating the running time as a function from the input.

C.Estimating the running time as a function from the memory size.

D.None of the above is correct.

085: Which of the following is the best reason forcing us to analyze algorithms? D

A.To save valuable resources.

B.To develop new and efficient algorithms.

C.To be able to compare among algorithms that can solve the same problem.

D.All of the above are correct.

086: Using the big O notations, which of the following is correct? D

A. $O(2^n) < O(n^2) < O(n^2 \log(n)) < O(n^3) < O(!n)$

B. $O(n) < O(n^2) < O(n^2 \log(n)) < O(en) < O(!n)$

C. $O(n) < O(n^2) < O(n^2 \log(n)) < O(3^n) < O(!n)$

D. $O(n) < O(n^2) < O(n^2 \log(n)) < O(n^3) < O(!n)$

088: What is the best complexity one can achieve for sorting a list of a partially sorted list?

D

A. $O(n)$.

B.something between $O(\log(n))$ and $O(n)$.

C.something between $O(n \log(n))$ and $O(n^2)$.

D.none of the above is correct.

089: Usually, Stacks are called FILO for which of the following reasons: C

A.the last element to “push” is the last element to “pop”.

B.the first element to “push” is the first element to “pop”.

C.the last element to “push” is the first element to “pop”.

D.none of the above is correct.

090: Dynamic memory allocation is recommended in which of the following situations: B

A.the size of the data is known to be (and remain) constant.

B.the size of the data is initially zero and expand (or shrink) over time.

C.there is an upper limit on the number of data items needed.

D.none of the above is correct.

091: Dynamic memory allocation can be implemented using: D

A.Arrays.

B.Singly Linked Lists.

C.Doubly Linked Lists.

D.both B and C.

092: What is the correct code for inserting element with value X after the element pointed to by P in a Singly Linked List (SLL)? C

A.P.next = new Node(X, null);

B.P.next = new Node(X, P.next);

C.P.next = new Node(X, P.next); if (P == Tail) Tail = Tail.next;

D.None of the above is correct.

093: Which one from the following is classified as a non-linear data structure? D

A.Linked Lists.

B.Dynamic Stacks.

C.Circular Queues.

D.Binary Trees.

094: Usually, Stacks are called LIFO for which of the following reasons? C

A.The last element to “push” is the last element to “pop”.

B.The first element to “push” is the first element to “pop”.

C.The last element to “push” is the first element to “pop”.

D.None of the above is correct.

095: Dynamic stacks are characterized by which of the following features? A

A.It is a constrained version from the singly linked lists.

B.Limited to a predefined max stack size parameter.

C.Must leave one empty location to work correctly.

D.All of the above are correct.

096: Usually, Queues are called FIFO for which of the following reasons? B

A.The first element to “enQueue” is the last element to “deQueue”.

B.The first element to “enQueue” is the first element to “deQueue”.

C.The last element to “enQueue” is the first element to “deQueue”.

D.None of the above is correct.

098: In the static implementation of a circular queue with f and r indices, which of the following is correct? B

A.index f points to the front element and index r points to the rear element.

B.index f points to the front element and index r points to location immediately after the rear element.

C.index f points to the front element and index r points to the right-most location of the array.

D.none of the above is correct.

099: In a static circular queue with max queue size N = 17, if r = 4 and f = 11 then the queue size is: A

A.10

B.13

C.15

D.17

100: A binary tree is balanced if: C

A.all levels are full.

B.all nodes in the binary tree have both left and right child.

C.all levels above the lowest level are full.

D.none of the above is correct.

101: Which of the following is the correct procedure for deleting a node with non-empty child in a Binary Search Tree (BST): B

A.remove the node by making its parent points to the left child of the original node

B.remove the node by replacing it with its right-left most leaf node

C.remove the node by making its parent points to the right child of the original node

D.only disconnect the node from its parent

102: If the pre-order traverse of a Binary Search Tree (BST) is “9 5 3 7 20 17 15 18 25 23 28” then the inorder traverse is: B

A.28 23 25 18 15 17 20 7 3 5 9

B.3 5 7 9 15 17 18 20 23 25 28

C.3 7 5 15 18 17 23 28 25 20 9

D.none of the above is correct

103: Which of the following is a suitable application for the queue data structure? A

A.The printing jobs in a multitasking operating system.

B.Function calls and recursions.

C.Transforming “infix” to “postfix” expressions.

D.None of the above is correct.

104: The high of a binary tree is defined to be: C

A.the number of levels above the lowest level.

B. 2^{n-1} , where n is the maximum number of levels.

C.the depth of its deepest node.

D.none of the above is correct.

105: What is the complexity of the following sorting algorithm written in Java? C

A. $O(\log(n))$

B. $O(n \log(n))$

C. $O(n) \leq \text{Complexity} \leq O(n^2)$

D. $O(n^2)$

```
public static void Sort( int a[] ) {
    for ( int pass = 0; pass < a.length; pass++ ) {
        boolean F = false;
        for ( i = 0; i < a.length - 1; i++ )
            if ( a[i] > a[i+1] ) { int temp = a[i]; a[i] = a[i+1]; a[i+1] = temp; F = true }
        if ( !F ) break;
    }
}
```

106: What is wrong with the following pop() operation in the static stack class? C

A.Will not work correctly if the stack is empty.

B.The operation should be declared as private instead of public.

C.Always return the top element of the stack without affecting its contents.

D.Nothing is wrong and the operation will work correctly.

```
01  public int pop() {  
02      if (isEmpty()) {  
03          System.out.println("Stack is Empty"); return -1;  
04      }  
05      else { return S[t]; t = t - 1; }  
06  }
```

107: What is wrong with the following addToTail operation in the SinglyLinkedList Class? D

A.The Head pointer is not set correctly.

B.Nothing is wrong and the operation will work correctly.

C.The new Node(el) doesn't specify a next pointer value.

D.None of the above is correct.

```
public void addToTail( int el ) {  
    if ( isEmpty() ) head = tail = new Node( el );  
    else tail.next = new Node(el);  
}
```

108: Dynamic Queues can best be implemented with which of the following elementary data structures? A

A.Singly Linked Lists.

B.Binary Search Trees.

C.Single-Dimension arrays.

D.None of the above is correct.

109: If the pre-order traverse of a BST is “9 5 3 7 20 17 15 18 25 23 28” then the post-order traverse is: C

A.28 23 25 18 15 17 20 7 3 5 9

B.3 5 7 9 15 17 18 20 23 25 28

C.3 7 5 15 18 17 23 28 25 20 9

D.none of the above is correct.

110: If the pre-order traverse of a BST is “9 5 3 7 20 17 15 18 25 23 28” then: C

A.the binary tree is not balanced.

B.the binary tree is not balanced and its height is 4.

C.the binary tree is balanced and its height is 3.

D.none of the above is correct.

113: While transforming infix to postfix, if the top of the stack is “/” and the next token is “)”, the actions is: D

A.push the “)” token on the stack.

B.a pop operation is performed, placing the “/” in the output, then push the “)” token on the stack.

C.a pop operation is performed, placing the “/” in the output.

D.none of the above is correct.

114: While evaluating a postfix expression, if the top of the stack is “A” and the next token is “B”, the action is: A

A.push the token “B” on the stack.

B.a pop operation is performed, then the current operation is performed on “A” & “B”.

C.two pop operations are performed, then the current operation is performed on “A” & “B”.

D.none of the above is correct.

115: Dynamic Queues are characterized by which of the following features: D

A.it is a constrained version from a single dimension array.

B.limited to a predefined max queue size parameter.

C.must leave one empty location to work correctly.

D.none of the above is correct.

116: While evaluating a postfix expression, if the next token is “+”, the action is: C

- A.push the token.
- B.pop the last two elements from the stack and perform the “+” operation.
- C.pop the last two elements from the stack and perform the “+” operation, then push the results.
- D.none of the above is correct.

118: In a static circular queue, when incrementing f and r, the modulo N operation is used for what reason? C

- A.To be able to enforce priorities in the queue.
- B.To limit the size of the queue to the value N.
- C.To make f and r circulate through the array representing the queue.
- D.None of the above is correct.

120: Which of the following is a valid statements about circular singly linked list? D

- A.Two pointers, head and tail, must be defined and used.
- B.The next pointer of the last element in the list points to “NULL”.
- C.We can move through the list clockwise and counter clockwise without any overhead.
- D.None of the above is correct.

123: A starvation-free job-scheduling policy guarantees that no job waits indefinitely for service. Which of the following job-scheduling policies is starvation-free? A

- A.Round-Robin
- B.Priority Queuing
- C.Shortest Job First
- D.none of the above is correct

124: if x is a string then x^r denotes the reversal of x. if x and y are strings, then $(xy)^r =$ D

- A. xy^r
- B. yx^r
- C. $y^r x$
- D. $y^r x^r$

125: The object-oriented paradigm includes which of the following properties? D

- A.Encapsulation
- B.Inheritance
- C.Recursion
- D.both A & B are correct

126: Which of the following algorithms has running time $O(n^2)$ in the worst case but $O(n \log n)$ on average? C

- A.Linear Sort
- B.Merge Sort
- C.Quick Sort
- D.Heap Sort

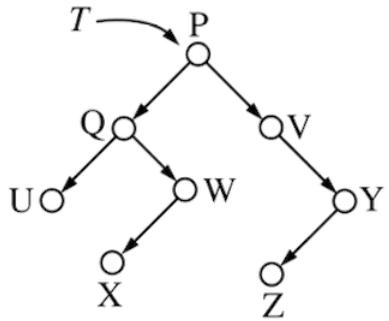
127: Which of the following is the name of the data structure in a compiler that is responsible for managing information about variables and their attributes? B

- A.Abstract Syntax Tree (AST)
- B.Symbol Table
- C.Attribute Grammar
- D.Parse Table

129: Which of the following represents a postorder traversal of the binary T shown below?

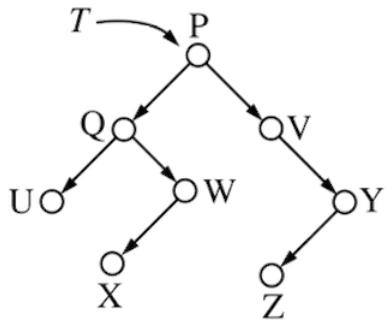
A

- A.U X W Q Z Y V P
- B.P Q U W X V Y Z
- C.U Q X W P V Z Y
- D.U X Z Q W Y V P



130: If the binary tree T shown below is a Binary Search Tree (BST), which of the following nodes contains the fourth smallest element in T? C

- A.the Q node
- B.the V node
- C.the W node
- D.the X node



132: Hash tables can contribute to an efficient average-case solution for all of the problems described below EXCEPT: A

- A.Range search: Given values a and b, find all the records whose key value is in the range a, b
- B.Counting distinct values: Given a set of n keys, determine the number of distinct key values
- C.Dynamic dictionary: Support the operations of insert, delete, and search in a dictionary
- D.Symbol table lookup: Given a program identifier, find its type and address

133: Let $G = (V, E)$ be a finite directed acyclic graph with $|E| > 0$. Which of the following must be true? D

- A.G has a vertex with no incoming edge
- B.G has a vertex with no outgoing edge
- C.G has an isolated vertex, that is, one with neither an incoming edge nor an outgoing edge
- D.both A & B are correct

135: Mergesort works by splitting a list of n numbers in half, sorting each half recursively, and merging the two halves. Which of the following data structures will allow mergesort to work in $O(n \log(n))$ time? D

- A.A singly linked list
- B.A doubly linked list
- C.An array
- D.all of the above are correct

137: Consider the 3 data structures for a set of n distinct integers shown below. For which of these data structures is the number of steps needed to find and remove the 7th largest element $O(\log n)$ in the worst case? D

- A.I only
 - B.III only
 - C.I & II
 - D.II & III
- I. A min-heap
 - II. An array of length n sorted in increasing order
 - III. A balanced binary search tree

138: Which of the following traversal techniques lists the nodes of a binary search tree in ascending order ? B

- A.Post-order traverse
- B.In-order traverse

C.Pre-order traverse

D.none of the above is correct

139: the concept of the Big O notation is important because: A

A.it can be used to decide the best algorithm that solves a given problem

B.it determines the maximum size of a problem that can be solved in a given system in amount of time

C.it is the lower bound of the growth rate of the algorithm

D.all of the above are correct

140: Which of the following algorithm design technique is used in the quick sort algorithm?

C

A.Dynamic Programming

B.Backtracking

C.Divide and Conquer

D.Greedy Method

141: Which of the following is true? B

A.The pop operation on a stack is considerably simpler than the remove operation on a queue

B.The contents of a queue can wrap around, while those of a stack cannot

C.The top of a stack corresponds to the front of a queue

D.none of the above is correct

142: The postfix expression for the infix expression $A + B * (C + D) / F + D * E$ is : B

A. $AB + CD + * F / D + E *$

B. $ABCD + * F / + DE * +$

C. $A * B + CD / F * DE + +$

D. $A + * BCD / F * DE + +$

143: Which of the following uses FIFO method? A

A.Queue

B.Stack

C.Binary Search Tree

D.all of the above are correct

144: A circular linked list can be used for: C

A.Stack

B.Queue

C.Both Stack & Queue

D.Neither Stack nor Queue

145: Binary search tree has best case run-time complexity of $O(\log n)$. What is the worst case? B

A. $O(n^2)$

B. $O(n)$

C. $O(n \log n)$

D.none of the above is correct

146: An algorithm is: A

A.a step by step procedure to solve problem

B.a piece of code to be executed

C.a loosely written code to make final code

D.all of the above are correct

147: The complexity to merge two sorted lists of size m and n is : C

A. $O(m / n)$

B. $O(m \log n)$

C. $O(m + n)$

D.none of the above is correct

148: A queue data structure can be used for : B

A.expression parsing

B.resource allocation

C.recursion

D.all of the above are correct

149: Which of the following algorithm cannot be designed without recursion : D

A.Tower of Hanoi

B.Fibonacci Series

C.Tree Traversal

D.none of the above is correct

150: Static Circular Queues are characterized by which of the following features: A

A.It is a constrained version from a single dimension array

B.Limited to a predefined max queue size parameter

C.Must leave one empty location to work correctly

D.All of the above are correct

151: which of the following array elements are not heap? B

A.15 10 14 8 6 12 13 1 2 5 3 9 7 11 4

B.15 12 14 11 13 9 8 7 6 5 4 1 10 2 3

C.15 14 10 13 12 6 8 4 11 7 9 3 5 2 1

D.15 14 13 11 12 10 8 9 3 7 5 6 4 2 1

152: If the pivot for the Quick Sort algorithm is selected to be the largest element in every iteration, then the complexity is: D

A. $O(n \log(n))$

B. $O(n \log(n)) \leq QS \leq O(n^2)$

C. $O(n^2 \log(n))$

D.none of the above is correct

153: The complexity of the Heap Sort (HS) algorithm is: C

A. $O(n^2)$

B. $O(n \log(n)) \leq HS \leq O(n^2)$

C. $O(n \log(n))$

D.none of the above is correct

182: Which of the following applications may use a stack? D

A.A parentheses balancing program

B.Functions call and return

C.Compiler Syntax Analyzer

D.all of the above are correct

185: Depth First Search is equivalent to which of the traversal in the Binary Trees? A

A.Pre-order Traversal

B.Post-order Traversal

C.In-order Traversal

D.none of the above is correct

186: Time Complexity of Depth First Search (DFS) is? (V = number of vertices, E = number of edges) A

A. $O(V + E)$

B. $O(V)$

C. $O(E)$

D.none of the above is correct

187: The Data structure used in standard implementation of Depth First Search is? A

A.Stack

B.Queue

C.Linked List

D.none of the above is correct

188: The Depth First Search (DFS) traversal of a graph will result into? B

A.Linked List

B.Tree

C.Graph with back edges

D.none of the above is correct

190: What can be the applications of Depth First Search? D

A.For generating topological sort of a graph

B.For generating Strongly Connected Components of a directed graph

C.Detecting cycles in the graph

D.all of the above are correct

191: When the Depth First Search of a graph is unique? B

A.When the graph is a Binary Tree

B.When the graph is a Linked List

C.When the graph is a n-ary Tree

D.none of the above is correct

192: Regarding implementation of Depth First Search using stacks, what is the maximum distance between two nodes present in the stack? (considering each edge length 1) A

A.Can be anything

B.0

C.At most 1

D.Insufficient Information

193: In Depth First Search, how many times a node is visited? C

A.Once

B.Twice

C.Equivalent to number of indegree of the node

D.none of the above is correct

194: Breadth First Search is equivalent to which of the traversals in the Binary Trees? C

A.Pre-order Traversal

B.Post-order Traversal

C.Level-order Traversal

D.In-order Traversal

195: Time Complexity of Breadth First Search is? (V = number of vertices, E = number of edges) A

A. $O(V + E)$

B. $O(V)$

C. $O(E)$

D.none of the above is correct

196: The Data structure used in standard implementation of Breadth First Search is? B

A.Stack

B.Queue

C.Linked List

D.none of the above is correct

197: The Breadth First Search traversal of a graph will result into? B

A.Linked List

B.Tree

C.Graph with back edges

D.all of the above are correct

198: A person wants to visit some places. He starts from a vertex and then wants to visit every place connected to this vertex and so on. What algorithm he should use? B

A.Depth First Search

B.Breadth First Search

C.Trim's Algorithm

D.none of the above is correct

199: What can be the applications of Breadth First Search? D

A.Finding shortest path between two nodes

B.Finding bipartiteness of a graph

C.GPS navigation system

D.all of the above are correct

200: When the Breadth First Search of a graph is unique? B

A.When the graph is a Binary Tree

B.When the graph is a Linked List

C.When the graph is a n-ary Tree

D.none of the above is correct

201: Regarding implementation of Breadth First Search using queues, what is the maximum distance between two nodes present in the queue? (considering each edge length 1) C

A.Can be anything

B.0

C.At most 1

D.Insufficient Information

202: In BFS, how many times a node is visited? C

A.Once

B.Twice

C.equivalent to number of indegree of the node

D.none of the above is correct

204: A program checks spelling in the following way: a hash table has been defined in which each entry is a Boolean variable initialized to false; a hash function has been applied to each word in the dictionary and the appropriate entry in the has table has been set to true; to check the spelling in a document, the hash function is applied to every word in the document, and the appropriate entry in the hash table is examined. Which of the following is correct ? B

A.true means the word was in the dictionary

B.false means the word was not in the dictionary

C.the hash table size should increase with document size

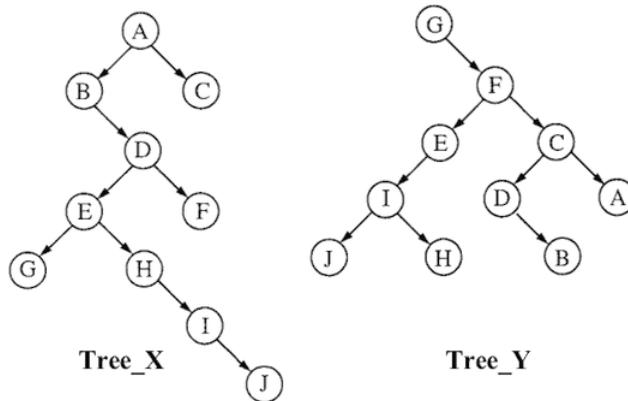
D.all of the above are correct

205: A certain algorithm A has been shown to have running time $O(n^{2.5})$, where n is the size of the input. Which of the following is NOT true about algorithm A? C

- A.For all N, there may be some inputs for which the running time is less than $n^{2.6}$ seconds
- B.For all N, there may be some inputs for which the running time is less than $n^{2.4}$ seconds
- C.For all N, there may be some inputs for which the running time is more than $n^{2.6}$ seconds
- D.For all N, there may be some inputs for which the running time is more than $n^{2.4}$ seconds

206: For Tree_X and Tree_Y shown below, which traversals will produce the same sequence of node names? B

- A.Preorder, Postorder
- B.Postorder, Inorder
- C.Postorder, Postorder
- D.Postorder, Preorder



207: Consider the following sorting algorithms, which has a running time that is dependent on the initial ordering of the input? D

- A.Quick Sort
- B.Merge Sort
- C.Bubble Sort

D.all of the above are correct

208: Consider the code given below, which of the following best describes the complexity of $f(x)$ as a function of x ? A

A.Exponential

B.Linear

C.Logarithmic

D.Quadratic

```
int f(int x)
{ if (x < 1) return 1 ;
  else return f(x - 1) + g(x) ;
}

int g(int x)
{ if (x < 2) return 1 ;
  else return f(x - 1) + g(x/2) ;
}
```

209: Let $G = (V, E)$ be a connected, undirected graph, and let a and b be two distinct vertices in V . Let $P1$ be the problem of finding a shortest simple path between a and b , and let $P2$ be the problem of finding a longest simple path between a and b . Which of the following statements about $P1$ and $P2$ is true? B

A.Both $P1$ and $P2$ can be solved in polynomial time

B. $P1$ can be solved in polynomial time but $P2$ is not known to be solvable in polynomial time

C. $P1$ not known to be solvable in polynomial time but $P2$ can be solved in polynomial time

D.none of the above is correct

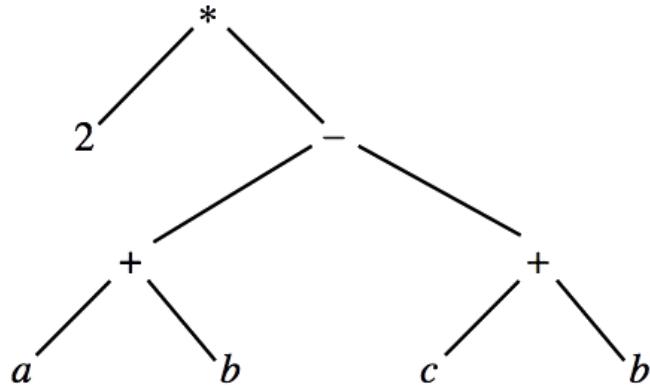
210: Which of the following arithmetic expressions corresponds to the parse tree given by the below diagram? B

A. $2(a + b - c + b)$

B. $2((a + b) - (c + b))$

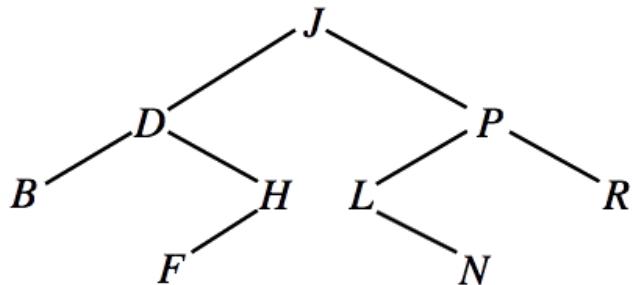
C. $2(a + b) - 2(c + b)$

D.none of the above is correct



211: The binary tree below can be used to store a sorted list so that an inorder tree walk generates the list in order. If the new entry K is to be made in such a way that alphabetical order is preserved, where should it be inserted? A

- A.As left child of L
- B.As left child of N
- C.As right child of H
- D.As right child of R



229: Which of the following is true concerning the complexity of the bubble sort, the quick sort, and the heap sort? C

- A.all can achieve $O(n \log(n))$ generally
- B.all have $O(n^2)$ worst case
- C.only the bubble sort can achieve $O(n)$ if the list is already sorted

D.only the heap sort can achieve $O(n)$ if the list is already sorted

230: What is wrong with the following method that is part of the Java SinglyLinkedList class definition? B

A.will not run due to a general compiler error

B.a compiler error due to object myStack has not been defined

C.will run with a run time error hanging the computer

D.nothing is wrong and the method will work correctly

```
public void printListReverse() {  
    Node tmp = head;  
    System.out.println("The Singly Linked List in Reverse Order is:");  
    while (tmp != null) {  
        myStack.push(tmp.info);  
        tmp = tmp.next;  
    }  
    while (!myStack.isEmpty()) {  
        System.out.println(myStack.pop());  
    }  
}
```

297: In a stack, if a user tries to remove an element from empty stack it is called A

A.Underflow

B.Empty collection

C.Overflow

D.Garbage Collection

298: Pushing an element into stack already having five elements and stack size of 5, then stack becomes A

A.Overflow

B.Crash

C.Underflow

D.User flow

299: Which of the following applications do not need a stack? D

- A.A parentheses balancing program
- B.Tracking of local variables at run time
- C.Compiler Syntax Analyzer
- D.Data Transfer between two asynchronous process

300: Consider the usual algorithm for determining whether a sequence of parentheses is balanced. The maximum number of parentheses that appear on the stack AT ANY ONE TIME when the algorithm analyzes: (())(()) are: C

- A.1
- B.2
- C.3
- D.4 or more

301: Consider the usual algorithm for determining whether a sequence of parentheses is balanced. Suppose that you run the algorithm on a sequence that contains 2 left parentheses and 3 right parentheses (in some order). The maximum number of parentheses that appear on the stack AT ANY ONE TIME during the computation? B

- A.1
- B.2
- C.3
- D.4 or more

302: What is the value of the postfix expression 6 3 2 4 + - *: D

- A.1
- B.40
- C.74
- D.-18

303: Here is an infix expression: $4 + 3 * (6 * 3 - 12)$. Suppose that we are using the usual stack algorithm to convert the expression from infix to postfix notation. The maximum number of symbols that will appear on the stack AT ONE TIME during the conversion of this expression? D

- A.1

B.2

C.3

D.4

304: The postfix form of the expression $(A + B) * (C * D - E) * F / G$ is : C

A. $A B + C D * E - F G / **$

B. $A B + C D * E - F * * G /$

C. $A B + C D * E - * F * G /$

D. $A B + C D E * - * F * G /$

305: The data structure required to check whether an expression contains balanced parenthesis is? A

A.Stack

B.Queue

C.Array

D.Tree

306: What data structure would you mostly likely see in a non recursive implementation of a recursive algorithm? B

A.Linked List

B.Stack

C.Queue

D.Tree

307: Which data structure is needed to convert infix notation to postfix notation? D

A.Branch

B.Tree

C.Queue

D.Stack

308: The prefix form of an infix expression $(p + q) - (r * t)$ is? C

A. $+ p q - * r t$

B.- +pqr * t

C.- +pq * rt

D.- + * pqrt

309: Which one is the equivalent postfix expression for the infix expression : $(A + B ^ D) / (E - F) + G A$

A.(A B D ^ + E F - / G +)

B.(A B D + ^ E F - / G +)

C.(A B D ^ + E F/- G +)

D.(A B D E F + ^ / - G +)

310: Which of the following statement(s) about stack data structure is/are NOT correct? C

A.Linked List are used for implementing Stacks

B.Top of the Stack always contain the new node

C.Stack is the FIFO data structure

D.Null link is present in the last node at the bottom of the stack

311: A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as a ? A

A.Queue

B.Stack

C.Tree

D.Linked List

312: The data structure required for Breadth First Traversal on a graph is? C

A.Stack

B.Array

C.Queue

D.Tree

313: A normal queue, if implemented using an array of size MAX_SIZE, gets full when A

A.Rear = MAX_SIZE - 1

B.Front = (rear + 1)mod MAX_SIZE

C.Front = rear + 1

D.Rear = front

314: Consider an implementation of unsorted singly linked list. Suppose it has its representation with a head pointer only. Given the representation, which of the following operation can be implemented in O(1) time?

- i) Insertion at the front of the linked list
- ii) Insertion at the end of the linked list
- iii) Deletion of the front node of the linked list
- iv) Deletion of the last node of the linked list

A.I and II

B.I and III

C.I, II and III

D.I, II and IV

315: In linked list each node contain minimum of two fields. One field is data field to store the data second field is? C

A.Pointer to character

B.Pointer to integer

C.Pointer to node

D.Node

316: What would be the time complexity to insert an element at the front of the linked list (head is known)? A

A.O(1)

B.O(n)

C.O(log n)

D.O(n log n)

317: What would be the time complexity to find an element in the linked list? C

A.O(1)

B.O(log n)

C.O(n)

D.O($n \log n$)

318: What would be the time complexity to insert an element at the second position in the linked list? A

A.O(1)

B.O($\log n$)

C.O(n)

D.O($n \log n$)

319: Linked lists are not suitable to for the implementation of? C

A.Bubble sort

B.Polynomial manipulation

C.Binary search

D.all of the above are correct

320: Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity? D

A.Insertion sort

B.Quick sort

C.Heap sort

D.Merge sort

321: Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list? A

A.Possible if X is not last node

B.Possible if size of linked list is even

C.Possible if size of linked list is odd

D.Possible if X is not first node

322: You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list? C

A.Delete the first element

B.Insert a new element as a first element

C.Delete the last element of the list

D.Add a new element at the end of the list

324: Which of the following is false about a doubly linked list? D

A.We can navigate in both the directions

B.It requires more space than a singly linked list

C.The insertion and deletion of a node take a bit longer

D.Implementing a doubly linked list is easier than singly linked list

325: What is the time complexity of inserting a node in a doubly linked list? C

A. $O(n \log n)$

B. $O(\log n)$

C. $O(n)$

D. $O(1)$

326: What differentiates a circular linked list from a normal linked list? C

A.You cannot have the next pointer point to null in a circular linked list

B.It is faster to traverse the circular linked list

C.You may or may not have the next pointer point to null in a circular linked list

D.Head node is known in circular linked list

327: Which of the following application makes use of a circular linked list? C

A.Undo operation in a text editor

B.Recursive function calls

C.Allocating CPU to resources

D.Implement Hash Tables

328: Which of the following is false about a circular linked list? B

A.Every node has a successor

B.Time complexity of inserting a new node at the head of the list is $O(1)$

C.Time complexity for deleting the last node is $O(n)$

D.We can traverse the whole circular linked list by starting from any point

329: Consider a small circular linked list. How to detect the presence of cycles in this list effectively? B

A.Keep one node as head and traverse another temp node till the end to check if its 'next' points to head

B.Have fast and slow pointers with the fast advancing two nodes at a time and slow advancing by one node at a time

C.Cannot determine, you have to pre-define if the list contains cycles

D.Circular linked list itself represents a cycle. So no new cycles cannot be generated

330: Minimum number of queues to implement a stack is : C

A.3

B.4

C.1

D.2

331: What is the time complexity to insert a node based on key in a priority queue? C

A. $O(n \log n)$

B. $O(\log n)$

C. $O(n)$

D. $O(1)$

332: What is not a disadvantage of priority scheduling in operating systems? C

A.A low priority process might have to wait indefinitely for the CPU

B.If the system crashes, the low priority systems may be lost permanently

C.Interrupt handling

D.Indefinite blocking

333: Which of the following is not an advantage of priority queue? D

A.Easy to implement

B.Processes with different priority can be efficiently handled

C.Applications with differing requirements

D.Easy to delete elements in any case

334: Which of the following is not the method to represent Sparse Matrix? C

A.Doubly Linked List

B.Singly Linked List

C.Heaps

D.Arrays

335: What is a hash table? B

A.A **structure that maps values to keys**

B.A structure that maps keys to values

C.A structure used for storage

D.A structure used to implement stack and queue

336: If several elements are competing for the same bucket in the hash table, what is it called? C

A.Diffusion

B.Replication

C.Collision

D.Duplication

337: What is a hash function? B

A.A function has allocated memory to keys

B.A function that computes the location of the key in the array

C.A function that creates an array

D.A function that computes the location of the values in the array

338: Which of the following is not a technique to avoid a collision? D

A.Make the hash function appear random

B.Use the chaining method

C.Use uniform hashing

D.Increasing hash table size

339: In simple chaining to resolve collisions in hashing, what data structure is appropriate?

B

A.Singly linked list

B.Doubly linked list

C.Circular linked list

D.Binary trees

378: A hash table of length 10 uses open addressing with hash function $h(k)=k \bmod 10$, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below. Which one of the following choices gives a possible order in which the key values could have been inserted in the table? C

A.46, 42, 34, 52, 23, 33

B.34, 42, 23, 52, 33, 46

C.46, 34, 42, 23, 52, 33

D.42, 46, 33, 23, 34, 52

0	1	2	3	4	5	6	7	8	9
		42	23	34	52	46	33		

379: How many different insertion sequences of the key values using the hash function $h(k) = k \bmod 10$ and linear probing will result in the hash table shown above? C

A.10

B.20

C.30

D.40

0	1	2	3	4	5	6	7	8	9
		42	23	34	52	46	33		

380: The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function $h(k) = k \bmod 10$ and linear probing. What is the resultant hash table? C

- A.{ Empty, Empty, 2, 23, Empty, 15, Empty, Empty, 18, Empty }
- B.{ Empty, Empty, 12, 13, Empty, 5, Empty, Empty, 18, Empty }
- C.{ Empty, Empty, 12, 13, 2, 3, 23, 5, 18, 15 }
- D.{ Empty, Empty, (12 2), (13 3 23), Empty, (5 15), Empty, Empty, 18, Empty }

382: Given the following input (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) and the hash function $x \bmod 10$, which of the following statements are true?

- i. 9679, 1989, 4199 hash to the same value
- ii. 1471, 6171 hash to the same value
- iii. All elements hash to the same value
- iv. Each element hashes to a different value C

- A.I only
- B.ii only
- C.I and ii only
- D.iii or iv

383: Consider a hash table with 100 slots. Collisions are resolved using chaining. Assuming simple uniform hashing, what is the probability that the first 3 slots are unfilled after the first 3 insertions? A

- A. $(97 \times 97 \times 97) / 100^3$
- B. $(99 \times 98 \times 97) / 100^3$
- C. $(97 \times 96 \times 95) / 100^3$
- D. $(97 \times 96 \times 95) / (3! \times 100^3)$

384: Which of the following statement(s) is TRUE?

1. A hash function takes a message of arbitrary length and generates a fixed length code
2. A hash function takes a message of fixed length and generates a code of variable length
3. A hash function may give the same hash value for distinct messages C

A.1 only

B.2 and 3 only

C.1 and 3 only

D.2 only

385: A hash function h defined $h(\text{key}) = \text{key} \bmod 7$, with linear probing, is used to insert the keys 44, 45, 79, 55, 91, 18, 63 into a table indexed from 0 to 6. What will be the location of key 18? C

A.3

B.4

C.5

D.6

386: An advantage of chained hash table (external hashing) over the open addressing scheme is C

A.Worst case complexity of search operations is less

B.Space used is less

C.Deletion is easier

D.None of the above is correct

387: Insert the characters of the string K R P C S N Y T J M into a hash table of size 10. Use the hash function : C

A.Y

B.C

C.M

D.P

$$h(x) = (\text{ord}(x) - \text{ord}("a") + 1) \bmod 10$$

388: A hash table with ten buckets with one slot per bucket is shown in the following figure. The symbols S1 to S7 initially entered using a hashing function with linear probing. The maximum number of comparisons needed in searching an item that is not present is : B

- A.4
- B.5
- C.6
- D.3

0	1	2	3	4	5	6	7	8	9
S7	S1		S4	S2		S5		S6	S3

389: Consider a hash table of size $m = 10000$, and the hash function $h(K) = \text{floor}(m(KA \bmod 1))$ for $A = (\sqrt{5} - 1)/2$. The key 123456 is mapped to location : B

- A.46
- B.41
- C.43
- D.48

390: Consider a 13 element hash table for which $f(\text{key}) = \text{key mod } 13$ is used with integer keys. Assuming linear probing is used for collision resolution, at which location would the key 103 be inserted, if the keys 661, 182, 24 and 103 are inserted in that order? B

- A.0
- B.1
- C.11

D.12

391: Which one of the following is an application of Queue Data Structure? D

- A.When a resource is shared among multiple consumers
- B.When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes
- C.Load Balancing
- D.All of the above are correct

392: How many stacks are needed to implement a queue. Consider the situation where no other data structure like arrays, linked list is available to you. B

- A.1
- B.2
- C.3
- D.4

393: A priority queue can efficiently implemented using which of the following data structures? Assume that the number of insert and peek (operation to see the current highest priority item) and extraction (remove the highest priority item) operations are almost same. C

- A.Array
- B.Linked List
- C.Heap Data Structures like Binary Heap
- D.None of the above is correct

395: Suppose a circular queue of capacity $(n - 1)$ elements is implemented with an array of n elements. Assume that the insertion and deletion operation are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = 0. The conditions to detect queue full and queue empty are A

- A.Full: $(\text{REAR}+1) \bmod n == \text{FRONT}$, empty: $\text{REAR} == \text{FRONT}$
- B.Full: $(\text{REAR}+1) \bmod n == \text{FRONT}$, empty: $(\text{FRONT}+1) \bmod n == \text{REAR}$
- C.Full: $\text{REAR} == \text{FRONT}$, empty: $(\text{REAR}+1) \bmod n == \text{FRONT}$
- D.Full: $(\text{FRONT}+1) \bmod n == \text{REAR}$, empty: $\text{REAR} == \text{FRONT}$

396: A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below: 10, 8, 5, 3, 2 Two new elements 1 and 7 are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is: D

- A.10, 8, 7, 5, 3, 2, 1
- B.10, 8, 7, 2, 3, 1, 5
- C.10, 8, 7, 1, 2, 3, 5
- D.10, 8, 7, 3, 2, 1, 5

397: Suppose implementation supports an instruction REVERSE, which reverses the order of elements on the stack, in addition to the PUSH and POP instructions. Which one of the following statements is TRUE with respect to this modified stack? C

- A.A queue cannot be implemented using this stack
- B.A queue can be implemented where ENQUEUE takes a single instruction and DEQUEUE takes a sequence of two instructions
- C.A queue can be implemented where ENQUEUE takes a sequence of three instructions and DEQUEUE takes a single instruction
- D.A queue can be implemented where both ENQUEUE and DEQUEUE take a single instruction each

398: A queue is implemented using an array such that ENQUEUE and DEQUEUE operations are performed efficiently. Which one of the following statements is CORRECT (n refers to the number of items in the queue)? A

- A.Both operations can be performed in O(1) time
- B.At most one operation can be performed in O(1) time but the worst case time for the other operation will be O(n)
- C.The worst case time complexity for both operations will be O(n)
- D.Worst case time complexity for both operations will be O(log n)

399: Which of the following option is not correct? C

- A.If the queue is implemented with a linked list, keeping track of a front pointer, Only rear pointer s will change during an insertion into an non-empty queue
- B.Queue data structure can be used to implement least recently used (LRU) page fault algorithm and Quick short algorithm

C.Queue data structure can be used to implement Quick short algorithm but not least recently used (LRU) page fault algorithm

D.Both (A) and (C) are correct

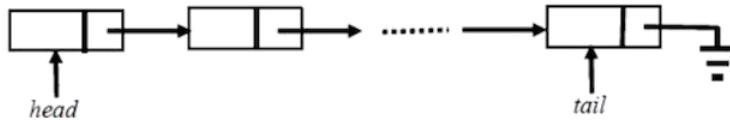
400: The below queue has head and tail pointers and n is the number of nodes. Enqueue is done by inserting a node at head and dequeue by deleting a node from tail. Which one is the time complexity of enqueue and dequeue? B

A. $O(1)$, $O(1)$

B. $O(1)$, $O(n)$

C. $O(n)$, $O(1)$

D. $O(n)$, $O(n)$



401: What is the worst case time complexity for search, insert and delete operations in a general Binary Search Tree? A

A. $O(n)$ for all

B. $O(\log n)$ for all

C. $O(\log n)$ for search and insert, and $O(n)$ for delete

D. $O(\log n)$ for search, and $O(n)$ for insert and delete

402: In delete operation of BST, we need inorder successor (or predecessor) of a node when the node to be deleted has both left and right child as non-empty. Which of the following is true about inorder successor needed in delete operation? B

A.Inorder Successor is always a leaf node

B.Inorder successor is always either a leaf node or a node with empty left child

C.Inorder successor may be an ancestor of the node

D.Inorder successor is always either a leaf node or a node with empty right child

403: How many distinct binary search trees can be created out of 4 distinct keys? B

A.4

B.14

C.24

D.42

404: Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the inorder traversal sequence of the resultant tree? C

A.7 5 1 0 3 2 4 6 8 9

B.0 2 4 3 1 6 5 9 8 7

C.0 1 2 3 4 5 6 7 8 9

D.9 8 6 4 2 3 0 1 5 7

405: The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, 16. What is the height of the binary search tree? B

A.2

B.3

C.4

D.6

406: The preorder traversal sequence of a binary search tree is 30, 20, 10, 15, 25, 23, 39, 35, 42. Which one of the following is the postorder traversal sequence of the same tree? D

A.10, 20, 15, 23, 25, 35, 42, 39, 30

B.15, 10, 25, 23, 20, 42, 35, 39, 30

C.15, 20, 10, 23, 25, 42, 35, 39, 30

D.15, 10, 23, 25, 20, 35, 42, 39, 30

407: Which of the following traversals is sufficient to construct BST from given traversals?

1) Inorder 2) Preorder 3) Postorder B

A.Any one of the given three traversals is sufficient

B.Either 2 or 3 is sufficient

C.2 and 3

D.1 and 3

408: While inserting the elements 71, 65, 84, 69, 67, 83 in an empty binary search tree (BST) in the sequence shown, the element in the lowest level is B

A.65

B.67

C.69

D.83

409: Suppose that we have numbers between 1 and 100 in a binary search tree and want to search for the number 55. Which of the following sequences CANNOT be the sequence of nodes examined? C

A.{10, 75, 64, 43, 60, 57, 55}

B.{90, 12, 68, 34, 62, 45, 55}

C.{9, 85, 47, 68, 43, 57, 55}

D.{79, 14, 72, 56, 16, 53, 55}

410: A binary search tree is used to locate the number 43. Which one of the following probe sequence is not possible? D

A.61, 52, 14, 17, 40, 43

B.10, 65, 31, 48, 37, 43

C.81, 61, 52, 14, 41, 43

D.17, 77, 27, 66, 18, 43

411: Access time of the symbolic table will be logarithmic if it is implemented by B

A.Linear list

B.Search tree

C.Hash table

D.Self organization list

412: What are the main applications of tree data structure?

- 1) Manipulate hierarchical data
- 2) Make information easy to search (see tree traversal)
- 3) Manipulate sorted lists of data

- 4) Router algorithms
- 5) Form of a multi-stage decision-making, like Chess Game
- 6) As a workflow for compositing digital images for visual effects D

A.1, 2, 3, 4 and 6

B.1, 2, 3, 4 and 5

C.1, 3, 4, 5 and 6

D.1, 2, 3, 4, 5 and 6

413: Level of a node is distance from root to that node. For example, level of root is 1 and levels of left and right children of root is 2. The maximum number of nodes on level i of a binary tree is : A

A. $2^{(i)-1}$

B. 2^i

C. $2^{(i+1)}$

D. $2^{[(i+1)/2]}$

414: In a complete k-ary tree, every internal node has exactly k children or no child. The number of leaves in such a tree with n internal nodes is: C

A. nk

B. $(n - 1) k + 1$

C. $n(k - 1) + 1$

D. $n(k - 1)$

415: The maximum number of binary trees that can be formed with three unlabeled nodes is: B

A.1

B.5

C.4

D.3

479: Which of the following points is/are true about Linked List data structure when it is compared with array D

A.It is easy to insert and delete elements in Linked List

- B.Random access is not allowed in a typical implementation of Linked Lists
- C.The size of array has to be pre-decided, linked lists can change their size any time
- D.All of the above are correct

480: In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is : D

A. $\log(n)$

B. $n/2$

C. $n \log(n)$

D.n

481: Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, cardinality will be the slowest? D

A.union only

B.intersection, membership

C.membership, cardinality

D.union, intersection

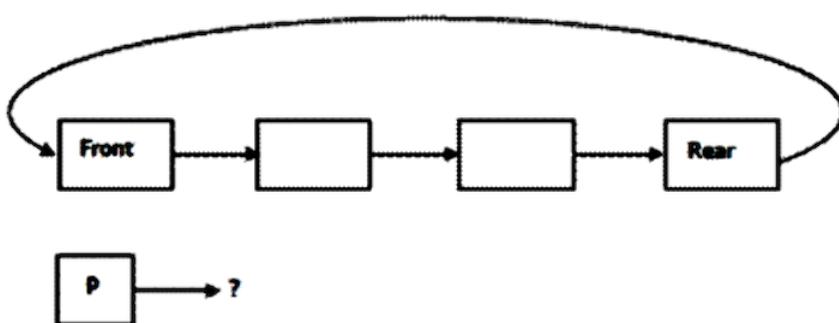
482: A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time? A

A.rear node

B.front node

C.not possible with a single pointer

D.node next to front



483: What are the time complexities of finding 8th element from beginning and 8th element from end in a singly linked list? Let n be the number of nodes in linked list, you may assume that $n > 8$. A

A.O(1) and O(n)

B.O(1) and O(1)

C.O(n) and O(1)

D.O(n) and O(n)

484: Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list? C

A.Possible if size of linked list is even

B.Possible if size of linked list is odd

C.Possible if X is not last node

D.Possible if X is not first node

485: You are given pointers to first and last nodes of a doubly linked list, which of the following operations are dependent on the length of the linked list? D

A.Delete the first element

B.Insert a new element as a first element

C.Delete the last element of the list

D.None of the above is correct

487: Let P be a singly linked list. Let Q be the pointer to an intermediate node x in the list. What is the worst-case time complexity of the best known algorithm to delete the node x from the list? C

A.O(n)

B.O(log n)

C.O(1)

D.None of the above is correct

488: Consider the following statements:

i. First-in-first out types of computations are efficiently supported by STACKS,

- ii. Implementing LISTS on linked lists is more efficient than implementing LISTS on an array for almost all the basic LIST operations,
 - iii. Implementing QUEUES on a circular array is more efficient than implementing QUEUES on a linear array with two indices,
 - iv. Last-in-first-out type of computations are efficiently supported by QUEUES.
- Which of the following is correct? A

A.(ii) and (iii) are true

B.(i) and (ii) are true

C.(iii) and (iv) are true

D.(ii) and (iv) are true

489: Suppose there are two singly linked lists both of which intersect at some point and become a single linked list. The head or start pointers of both the lists are known, but the intersecting node and lengths of lists are not known. What is worst case time complexity of optimal algorithm to find intersecting node from two intersecting linked lists? C

A. $O(n*m)$, where m, n are lengths of given lists

B. $O(n^2)$, where $m > n$ and m, n are lengths of given lists

C. $O(m+n)$, where m, n are lengths of given lists

D. $O(\min(m, n))$, where m, n are lengths of given lists

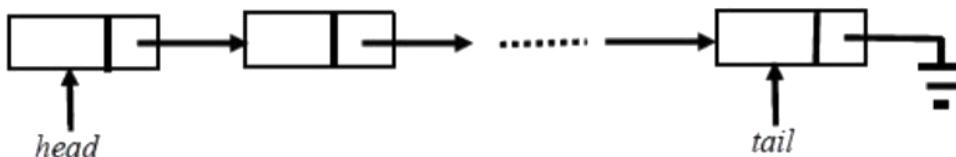
490: The below stack implementation has head and tail pointers. Let n denote the number of nodes in the stack. Let push be implemented by insert a new node at the head, and pop be implemented by delete a node from the head. Which one of the following is the time complexity of the most time-efficient implementation of push and pop for this data structure? A

A. $O(1), O(1)$

B. $O(1), O(n)$

C. $O(n), O(1)$

D. $O(n), O(n)$



491: In a doubly linked list, the number of pointers affected for an insertion operation will be : B

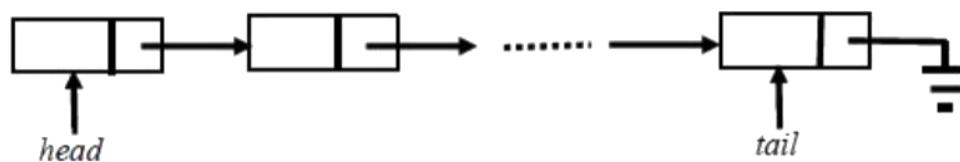
- A.1
- B.2 or 4
- C.4
- D.0

492: Consider an implementation of unsorted single linked list. Suppose it has its representation with a head and a tail pointers (i.e. pointers to the first and last nodes of the linked list). Which of the following operations can not be implemented in O(1) time? D

- A.Insertion at the front of the linked list
- B.Insertion at the end of the linked list
- C.Deletion of the front node of the linked list
- D.Deletion of the last node of the linked list

493: Consider a single linked list where head and tail are pointers to the first and last elements respectively of the linked list. The time for performing which of the given operations depends on the length of the linked list? C

- A.Delete the first element of the list
- B.Interchange the first two elements of the list
- C.Delete the last element of the list
- D.Add an element at the end of the list



495: The time required to search an element in a linked list of length n is B

- A.O ($\log n$)
- B.O (n)
- C.O (1)

D.O (n^2)

496: Which of the following operations is performed more efficiently by doubly linked list than by a singly linked list? A

- A.Deleting a node whose location is given
- B.Searching an unsorted list for a given item
- C.Inserting a node after the node with a given location
- D.Traversing the list to process each node in forward order

497: The minimum number of fields with each node of doubly linked list is C

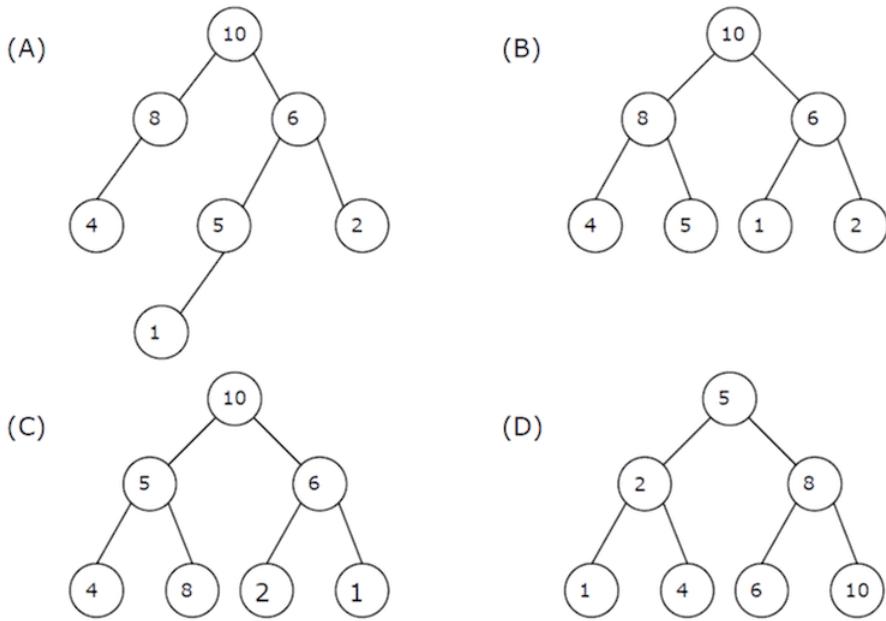
- A.1
- B.2
- C.3
- D.4

625: What is the time complexity of the Build Heap operation? Build Heap is used to build a binary heap from a given array which is needed in Heap Sort as a first step for sorting. A

- A.O($n \log n$)
- B.O(n^2)
- C.O($\log n$)
- D.O(n)

627: A max-heap is a heap where the value of each parent is greater than or equal to the values of its children. Which of the following is a max-heap? B

- A.(Figure A)
- B.(Figure B)
- C.(Figure C)
- D.(Figure D)



630: Consider a binary heap implemented using an array. Which one of the following array represents a binary heap? C

A.25,12,16,13,10,8,14

B.25,12,16,13,10,8,14

C.25,14,16,13,10,8,12

D.25,14,12,13,10,8,16

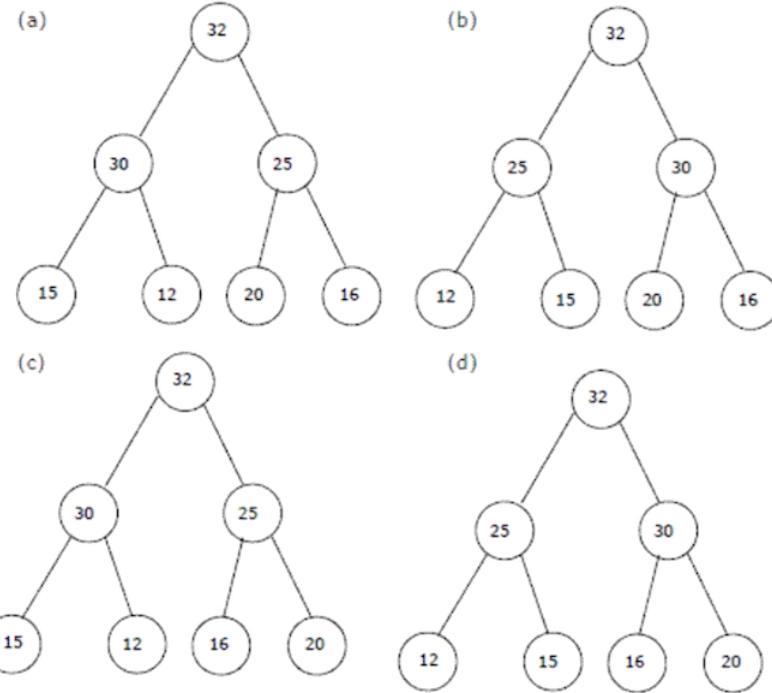
632: The elements 32, 15, 20, 30, 12, 25, 16 are inserted one by one in the given order into a Heap Tree. The resultant Heap is: A

A.Figure (a)

B.Figure (b)

C.Figure (c)

D.Figure (d)



637: Which of the following is true about the linked list implementation of a stack? D

- A.In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
- B.In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.
- C.Both A and B are correct.
- D.None of the above is correct.

644: Which of the following is TRUE about priority scheduling in operating systems? D

- A.High priority processes can take the CPU instantaneously
- B.The system can save its state in case of crashes
- C.Interrupt handling is possible
- D.All of the above are correct

652: What is the time complexity to insert a node based on key in a priority queue? C

- A. $O(n \log n)$

B.O (log n)

C.O (n)

D.O (n²)

653: What is not a disadvantage of priority scheduling in operating systems? C

A.A low priority process might have to wait indefinitely for the CPU.

B.If the system crashes, the low priority systems may be lost permanently.

C.Interrupt handling.

D.Indefinite blocking

654: Which of the following is not an advantage of a priority queue? D

A.Easy to implement.

B.Processes with different priority can be efficiently handled.

C.Applications with differing requirements.

D.Easy to delete elements in any case.

656: What is the time complexity of enqueue operation? D

A.O (log n)

B.O (n log n)

C.O (n)

D.O (1)

657: What is the need for a circular queue? A

A.Effective usage of memory.

B.Easier computations.

C.To delete elements based on priority.

D.All of the above are correct.

658: What is the best case time complexity of deleting a node in a Singly Linked list? D

A.O (n)

B.O (n²)

C.O (n log n)

D.O (1)

659: Which of the following statements are not correct with respect to Singly Linked List (SLL) and Doubly Linked List (DLL)? D

A.Complexity of Insertion and Deletion at known position is O(n) in SLL and O(1) in DLL.

B.SLL uses less memory per node than DLL.

C.DLL has more searching power than SLL.

D.Number of node fields in SLL is more than DLL.

689: What is the time complexity of searching for an element in a circular linked list? A

A.O(n)

B.O(n log n)

C.O(1)

D.None of the above is correct

690: The result of evaluating the postfix expression: 5 4 6 + * 4 9 3 / + * is B

A.700

B.350

C.500

D.None of the above is correct

691: Which of the following is the correct Postfix expression equivalent to the Infix expression $(A + B \wedge D)/(E - F) + G A$

A.A B D $\wedge + E F - / G +$

B.A B D $+ \wedge E F - / G +$

C.A B D $\wedge + E F / - G +$

D.None of the above is correct

692: Which of the following statement(s) about stack data structure is/are NOT correct? C

A.Linked List are used for implementing Stacks

B.Top of the Stack always contain the new node

C.Stack is the FIFO data structure

D.None of the above is correct

694: Which of the following is not an inherent application of stack? D

A.Reversing a string

B.Evaluation of postfix expression

C.Implementation of recursion

D.Job scheduling

117: In a static circular queue, location r must be kept empty for which of the following reasons? B

A.To speed up the enqueue operation.

B.To be able to identify the status when the queue is empty.

C.The initialization of the queue restricts us to do that.

D.None of the above is correct.

626: Suppose we are sorting an array of eight integers using heapsort, and we have just finished some heapify operations. The array now looks like this: 16 14 15 10 12 27 28. How many heapify operations have been performed? B

A.1

B.2

C.3 or 4

D.5 or 6

628: A 3-ary max heap is like a binary max heap, but instead of 2 children, nodes have 3 children. A 3-ary heap can be represented by an array as follows: The root is stored in the first location, a(0), nodes in the next level, from left to right, is stored from a(1) to a(3). The nodes from the second level of the tree from left to right are stored from a(4) location onward. An item x can be inserted into a 3-ary heap containing n items by placing x in the location a(n) and pushing it up the tree to satisfy the heap property. Which one of the following is a valid sequence of elements in an array representing 3-ary max heap? D

A.1, 3, 5, 6, 8, 9

B.9, 6, 3, 1, 8, 5

C.9, 3, 6, 8, 5, 1

D.9, 5, 6, 8, 3, 1

629: Suppose the elements 7, 2, 10 and 4 are inserted, in that order, into the valid 3- ary max heap found in the above question, Which one of the following is the sequence of items in the array representing the resultant heap? A

A.10, 7, 9, 8, 3, 1, 5, 2, 6, 4

B.10, 9, 8, 7, 6, 5, 4, 3, 2, 1

C.10, 9, 4, 5, 7, 6, 8, 2, 1, 3

D.10, 8, 6, 9, 7, 2, 3, 4, 1, 5

631: In a binary max heap containing n numbers, the smallest element can be found in time:
A

A.O(n)

B.O(log n)

C.O (n log n)

D.O (1)

636: Which one of the following is an application of the Stack Data Structure? D

A.Managing function calls.

B.The stock span problem.

C.Arithmetic expression evaluation.

D.All of the above are correct.

638: Consider the below pseudocode that uses a stack. What is the output for the input geeksquiz? B

A.geeksquizgeeksquiz

B.ziuqskeeg

C.geeksquiz

D.ziuqskeegziuqskeeg

```

declare a stack of characters
while ( there are more characters in the word to read )
{
    read a character
    push the character on the stack
}
while ( the stack is not empty )
{
    pop a character off the stack
    write the character to the screen
}

```

640: The following postfix expression with single digit operands is evaluated using a stack:
 $8\ 2\ 3\ ^\ / \ 2\ 3\ * + \ 5\ 1\ * -$ Where $^$ is the exponentiation operator. What are the top two elements of the stack after the first $*$ is evaluated? A

A.6, 1

B.5, 7

C.3, 2

D.1, 5

642: To evaluate an expression without any embedded function calls: A

A.one stack is enough.

B.two stacks are needed.

C.as many stacks as the height of the expression tree are needed.

D.none of the above is correct.

649: Consider the task of traversing a singly linked list in reverse order, i.e. from tail to head, what is the complexity of this operation for a singly linked list of size n? C

A. $O(1)$

B. $O(n)$

C. $O(n^2)$

D.None of the above is correct.

650: Which of the following is not an application of priority queue? C

A.Huffman codes

B.Interrupt handling in operating system

C.Undo operation in text editors

D.Bayesian spam filter

651: With what data structure can a priority queue be (best) implemented? C

A.Array

B.List

C.Heap

D.Tree

842: In a static circular queue with max queue size N, f and r indices, the queue size formula is: C

A. $(f - r) \bmod N$

B. $(N - f + r)$

C. $(N - f + r) \bmod N$

D.None of the above is correct

639: Below is an incorrect pseudocode for the algorithm which is supposed to determine whether a sequence of parentheses is balanced or not. Which of these unbalanced sequences does the above code think is balanced? A

A.((0)

B.0)(0)

C.(00))

D.(0))0

```
declare a character stack
while ( more input is available)
{
    read a character
    if ( the character is a '(' )
        push it on the stack
    else if ( the character is a ')' and the stack is not empty
        pop a character off the stack
    else
        print "unbalanced" and exit
}
print "balanced"
```

641: A single array A(1..MAXSIZE) is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (top1< top2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for STACK FULL is : D

A.(top1 = MAXSIZE/2) and (top2 = MAXSIZE/2+1)

B.top1 + top2 = MAXSIZE

C.(top1= MAXSIZE/2) or (top2 = MAXSIZE)

D.top1= top2 -1

695: Assume that the operators +, -, X are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, X, +, -. The postfix expression for the infix expression $a + b X c - d ^ e ^ f$ is? B

A.abc X+ def ^ ^ -

B.abc X+ de^f^ -

C.ab+c Xd - e ^f^

D.-+aXbc^ ^def