Data Structures Questions

UNIT 1: Introduction & ARRAYS

1. Define Data, Data Types and Data Structures.
2. Explain different types of Data Structures.
3. Why do we need a data structures?
4. What factor influence the choice of a particular Data Structures?
5. Differentiate between linear and non linear data structures.
6. Define algorithm. What are the steps involved in an algorithm?
7. Explain different ways of analyzing algorithm.
8. What do you mean by time complexity and space complexity of an algorithm?
9. Explain time space tradeoff.
10. Write an efficient algorithm to find the k^{th} element in the sequence of n elements.
11. Write the traversing algorithm for a linear array.
12. What do you mean by asymptotic notations? Explain various asymptotic notations.
13. Explain Big Oh notation in detail.
14. Explain the concept of best case, worst case and average case behavior of an algorithm with particular example.
15. Define abstract data types. Explain in brief.
16. What is row major order? Explain with example.
17. What is column major order? Explain with example.
18. State the merits and demerits of static and dynamic memory allocation techniques.
19. How two dimensional arrays are represented in memory? Also obtain the formula for calculating the address of any element stored in array (a) by row major order (b) by column major order. (Make necessary assumptions yourself).
20. Obtain addressing formula for an element in three dimensional array represented in column major order.
21. Obtain addressing formula for an element in three dimensional array represented in row major order.

22. Write a program in C to create an array of 10 elements. Take 10 elements in the array pass them to a function which prints the elements in reverse order.

23. Write an algorithm to multiply two matrices and determine complexity of the algorithm.

24. Given a 2-D array A[-100:100][-5:50]. Find the address of an element A[99, 49] considering the base address 10 and each element requires 4 bytes of storage. Follow row major order and column major order.

25. Each element of an array Data [20][50] requires 4 bytes of storage. Base address of array is 2000. Determine the location of Data[10][10] when the array is stored as: (a) Row major order (b) Column major order.

26. Write a C program to obtain the transpose of an nxn square matrix onto itself.

27. Write a C function to find a saddle point in 2-D matrix. (An mxn matrix is said to have a saddle point if some entry a[i][j] is the smallest value in row i and largest value in column j)

28. Write a program for merging of two unsorted arrays.

29. Write a program for merging of two sorted arrays.

30. What do you mean by complexity of an algorithm? Compute the worst case complexity for the following C code:

```
main ()
{
int s = 0, i, j, n;
        for(j=0; j<(3*n); j++)
        {
        for(i=0; i<n; i++)
        {
        s = s + i;
        }
        printf("%d",j);
        }
}
```

31. What do you understand by sparse matrices? Discuss its representation by giving suitable example.

LINKED LIST

1. What are the different types of the linked list? Give suitable examples of each.
2. Differentiate between array and linked list.
3. What is the advantage of linked list over array? Explain in brief.
4. Write an algorithm which reverses a singly linked list only by manipulating pointers.
5. Write a function which deletes the fifth node in doubly linked list.
6. Write a C function to insert new node at the beginning, at middle position and at the end of a singly linked list.
7. Write the difference between malloc and calloc functions. Why do we need dynamic memory allocation?
8. Write algorithm or C code to insert a node in a doubly linked list in the beginning.
9. How can we represent a polynomial in a linked list? Write an algorithm to add two polynomial represented by linked list.
10. What is generalized linked list? Explain the way of representation.
11. Give representation of the following generalized list:
    i. A = (1,2,(3,(4,5)),6)
    ii. A = (1,2,3,(4,(5,(6,7,8),9,( ),10,11,12)))
12. Write a program in C to delete a specific element in a singly linked list.
13. Doubly linked list takes more space than singly linked list for storing one extra address. In what condition could be a doubly linked list be more beneficial than singly linked list.
14. Write a C function that creates a new linear linked list by selecting alternatives elements of a given linear linked list.
15. Write a C function for inserting a new node at the end of a doubly linked list.
16. Write an algorithm to count the number of nodes between two given nodes in linked list.
17. Write an algorithm to find the location of an element in the given linked list. Is the binary search will be suitable for this search? Explain the reason.
18. What is doubly linked list? What are the advantages and disadvantages of doubly linked list?
19. Define circular linked list and circular doubly linked list by giving suitable examples.
20. Write an algorithm or C function that reverses order of all the elements in a singly linked list.
21. Write a C function that creates a new linear linked list by selecting alternative elements of a given linear linked list.

UNIT 2: STACK & QUEUE

STACK

1. Write an algorithm to evaluate a postfix expression. State the assumptions, if any, you make regarding the input.
2. Write the PUSH and POP functions in C simulating Push and Pop operations of STACK implemented using an array of integers.
3. Write the PUSH and POP functions in C simulating Push and Pop operations of STACK implemented using Linked List.
4. Convert the given Infix expression to Postfix expression using stack and show the detail of stack at each step of conversion.
   i. Expression: (a + b * c ^ d)*(e + f / g).
   ii. Expression: A + (B * C + D) / E
   iii. Expression: A – B / C + D * E + F
   iv. Expression: A * (B + C ^ D) – E ^ F * (G / H)
   v. Expression: A * (B + D) / E – F * (G + H / K)
   vi. Expression: ((A – (B + C) * D) / (E + F))
   vii. Expression: A + B ^ C ^ D – E * F / G
   Note: ^ indicates exponent operator.
5. What is tail recursion? Explain with example.
6. Write an algorithm to convert Infix notation to Postfix notation of an expression.
7. What is the Tower of Hanoi Problem? Give the solution in terms of disk move sequence on pegs A, B, C for 4 disks.
8. What is the Tower of Hanoi Problem? Give the solution in terms of disk move sequence on pegs A, B, C for 5 disks.
9. What is Stack? Implement stack with singly linked list.
10. Convert the following infix expression to Prefix expression using stack.
    i. Expression: ((2 + 3) * 4 + (5 * (6 + 7) * 8) + 9)
    ii. Expression: (A + B) * C – (D – E) ^ F
11. Give a recursive solution to Tower of Hanoi problem.
12. Give a data structure to implement two stacks in same array. Write C function to implement push operation on both the stacks.

13. Ackerman's function is defined as follows:

$A(m , n) = n + 1$ if $m = 0$

$A(m , n) = A(m – 1 , 1)$ if $m != 0 , n = 0$

$A(m , n) = A(m – 1 , A(m, n – 1))$ if $m != 0, n != 0$

Calculate the value for A(2,3). Show all intermediate steps of calculation.

14. Define the recursion. Write a recursive and non recursive program to calculate the factorial of a number.

15. Write algorithm to convert a postfix expression into an infix expression. Consider the following arithmetic expression in postfix notation:

7 5 2 + * 4 1 5 – / –

i. Find the value of the expression.

ii. Find the equivalent prefix form of the above expression.

16. Write a recursive program to find sum of digits of a number. Also calculate the time complexity.

17. Write down the applications of stack.

18. Evaluate the postfix expression: 8 2 – 4 + 5 6 7 – + *.

19. Translate the following string into polish notation and trace the content of stack

A – (B / C + (D % E * F) / G) * H

20. Explain the terms overflow and underflow with suitable example.

21. How we perform conversion of an expression from postfix to infix? Explain with suitable example.

22. How we perform conversion of an expression from postfix to prefix? Explain with suitable example.

23. Give the data structure to implement two stacks in same array. Write functions to implement push operations on both the stacks.

24. Write short notes on multiple stacks in an array.

25. Write a recursive program to compute greatest common divisor (GCD) of two integers.

26. If the Tower of Hanoi is operated on n = 10 disks, calculate the total number of moves.

QUEUE

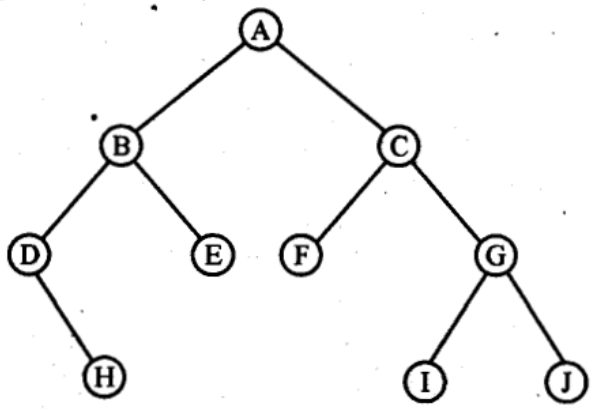
1. What is queue? What are the basic operations performed on it?
2. How do you implement a circular queue in C using array? Write routines to implement operations for it.
3. Differentiate between dequeue and priority queue.
4. Write down the applications of queue.
5. Write overflow conditions for Circular queue.
6. Explain why simple array implementation of queue is not practically useful.
7. Write C function for insertion, deletion and traversal of a linear queue.
8. What is circular queue? Write a C function to insert an element in circular queue.
9. Write a C function to delete element from circular queue implemented using array.
10. Define linear queue. State the advantages of circular queue over linear queue by giving suitable example.
11. Write an algorithm that reverses all elements in a queue.
12. What do you mean by DEQueue? Explain its types by suitable example
13. What do you mean by Priority Queue? Explain the types to maintain the priority queue.
14. Write a C program to implement a queue using linked list.
15. Write a program or function in C to find out duplicates elements in the queue.
16. A double ended-queue (dequeue) is a linear list in which additions may be made at either end. Obtain a data representation mapping a dequeue into one dimensional array. Write a C function to add and delete elements from either end of dequeue.

UNIT 3: TREES

1. Define tree, binary tree, complete binary tree and strict binary tree by giving suitable examples.
2. How a tree can be stored in the memory? Explain with example.
3. Following are the in-order and post-order traversal of a binary tree

   In order: D K I B A E G H J F C

   Post order: K D I E A G B F C J H
4. Write functions to implement recursing versions of pre-order, in-order and post-order traversals of a binary tree.
5. Write an algorithm or C function to implement binary tree insertion and deletion with example.
6. Write an algorithm or C function for various traversing techniques of binary tree with neat example.
7. Show that the maximum number of nodes in a binary tree of height h is $2^{h+1} - 1$.
8. If the in-order traversal of a binary tree is B, I, D, A, C, G, E, H, F and its post-order traversal is I, D, B, G, C, H, E, F, A. Determine the binary tree.
9. Write an algorithm to convert a forest into a binary tree.
10. What do you mean by threaded binary tree? Explain the types and traversing in a threaded binary tree.
11. How many null branches are there in a binary tree with 20 nodes?
12. Traverse the given tree using Pre-order, In-order and Post-order traversals.



13. Construct the binary tree given the following traversals:

    Pre-order:A, B, D, G, H, C, E, I, F

    In-order :G, D, H, B, A, E, I, C, F

14. Construct the binary tree given the following traversals:

Pre-order:F, A, E, K, C, D, H, G, B

In-order :E, A, C, K, F, H, D, B, G

15. If there are 27 nodes in a complete binary tree, what will be its height and how many nodes will be in the last level?

16. Construct an expression tree for the following algebraic expression:

(3a – b) ^ 2 (4c + 2d) ^ 3

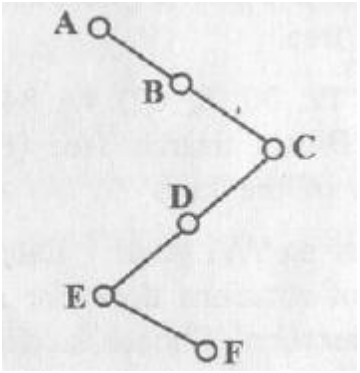
17. Write an algorithm to determine whether a binary tree is complete or not.

18. What is Huffman tree? Create a Huffman tree with the following numbers: 24, 55, 13, 67, 88, 36, 17, 61, 24, 76

19. Suppose characters a, b, c, d, e, f have probabilities 0.07, 0.09, 0.12, 0.22, 0.23, 0.27 respectively. Find an optimal Huffman code and draw the Huffman tree. What is the average code length?

20. Show that in any binary tree, the number of leaves nodes is one more than the number of nodes of degree two.

21. Write pre-order, in-order and post-order traversal of following binary tree.



22. Define 2-tree or extended binary tree.

23. The order of nodes in a binary tree in Pre-order and Post-order traversal are as follows:

Pre-order: 14, 4, 3, 9, 7, 5, 10, 15, 18, 16, 17, 20

Post-order: 3, 5, 7, 10, 9, 4, 17, 16, 20, 18, 15, 14

Draw the corresponding binary tree and write its in-order traversal.

24. Draw a Huffman tree for the following symbols whose frequency of occurrence in a message is stated along with the symbol below:

A: 15, B: 6, C:7, D:12, E:25, F:4, G:6, H:1, I:15

Decode the message: 1110100010111011