DATA STRUCTURE & ALGORITHMS
(CSEN 2101)

Time Allotted : 3 hrs **Full Marks : 70**

Figures out of the right margin indicate full marks.

Candidates are required to answer Group A and any 5 (five) from Group B to E, taking at least one from each group.

Candidates are required to give answer in their own words as far as practicable.

Group – A
(Multiple Choice Type Questions)

1. Choose the correct alternative for the following: **10 × 1 = 10**

(i) Reverse Polish notation is preferred over infix notation because
(a) the knowledge of precedence is not needed
(b) brackets will not be needed
(c) both of the above
(d) none of the above.

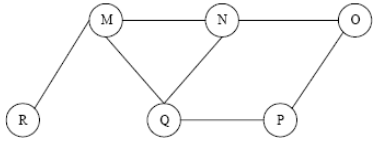
(ii) The following postfix expression with single digit operands is evaluated using a stack:
8 2 3 ^ / 2 3 * + 5 1 * -
Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are:
(a) 6,1 (b) 5,7 (c) 3,2 (d) 1,5.

(iii) Let n be the number of nodes in a linked list, you may assume that n > 8. What are the time complexities of finding 8th element from the beginning and 8th element from the end in that singly linked list?
(a) O(1) and O(n) (b) O(1) and O(1)
(c) O(n) and O(1) (d) O(n) and O(n).

(iv) Suppose a binary tree is constructed with n nodes, such that each node has exactly either zero or two children. The maximum height of the tree will be?
(a) (n + 1)/ 2 (b) (n - 1)/2
(c) n / 2 – 1 (d) (n + 1) / 2 – 1.

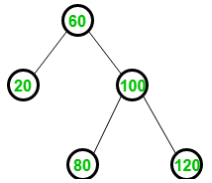
(v) Consider a binary max-heap implemented using an array. Which one of the following array represents a binary max-heap?
(a) 25,12,16,13,10,8,14 (b) 25,12,16,13,10,8,14
(c) 25,14,16,13,10,8,12 (d) 25,14,12,13,10,8,16.

(vi) The Breadth First Search algorithm has been implemented using the queue data structure. One possible order of visiting the nodes of the following graph is

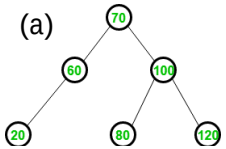


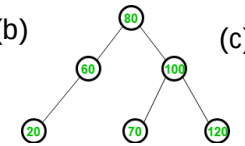
(a) MNOPQR (b)NQMPOR (c) QMNPRO (d) QMNPOR.

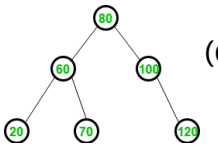
(vii) Consider the following AVL tree.



Which of the following is updated AVL tree after insertion of 70?

(a)


(b)


(c)


(d) None.

(viii) Running time of BUILD-MAX-HEAP function of Heap Sort is tightly bounded by which of the following?
(a) O(n) (b) O(n log n) (c) O(1) (d) None of these.

(ix) What will be the post order traversal of the given tree?

(a) 1, 2, 3, 4, 5, 6 (b) 4, 3, 6, 5, 2, 1
(c) 3, 4, 6, 5, 2, 1 (d) 5, 3, 4, 6, 2, 1

(c) It is said that in interpolation search, in each pass the search space is reduced to $n^{1/2}$ from n. Show that if the search ends in k passes, then $k = O(\log_2 \log_2 n)$.

(2 + 2) + 5 + 3=12

(x) Suppose you have the following hash table, implemented using linear probing. The hash function is h(x) =x mod m, where m is the size of the hash table.

0	1	2	3	4	5	6	7	8
9	18		12	3	14	4	21	

In which of the following order could the elements have been added to the hash table? Assume that the hash table has never been resized, and no elements have been deleted yet.

A. 9, 14, 4, 18, 12, 3, 21
B. 12, 3, 14, 18, 4, 9, 21
C. 12, 14, 3, 9, 4, 18, 21
D. 9, 12, 14, 3, 4, 21, 18
E. 12, 9, 18, 3, 14, 21, 4

(a) Only A (b) D & E (c) C & D (d) Only B.

Group – B

2. (a) State the advantages and disadvantages of linked list over array.

(b) Write a pseudo code or C function to perform enqueue in a circular queue, which is implemented by a circular linked list. What will be the time complexity of your code?

(c) If the number of nonzero elements in the triple representation of a sparse matrix of m × n dimension is K, then what is the time complexity to read an element from the index [i][j] of the sparse matrix, mind that you have stored the sparse matrix in triple format?

(d) Given a 2D matrix of dimensions 100 × 200 and given the base address of the 2D array is 2016, find out the address where the element with index (30, 40) is stored. Assume that the indices of the matrix start from (0, 0) and column major ordering is used to map the storage to memory.

3 + (4 + 1) + 2 + 2 = 12

3. (a) State if the following statement is correct and also explain your answer.
$O(n^2) = 2n^2+100$

(b) Why $n^{0.99}$ is asymptotically bigger than $(\log n)^{99}$?

(c) i) Suggest a pseudo code / C function to find the middle element of a single linked list in a single pass. (Please note that if the list has even number of elements (n) then middle element will be the first middle element ($\lfloor n/2 \rfloor$).

ii) What will be the complexity of the following code segment and why?

```
for (i = n; i > 0; i--)
{
  for(j = 0; j < i; j++)
  {
    printf ("DS\n");
  }
```

```
}
```

2 +2 + (6 + 2) =12

Group – C

4. (a) Why do we need sparse representation of matrices? State the name of the two representations.

(b)
```
void quicksort( int arr[ ], int low, int high)
{
    if(high > low)
    {
      int  mid = partition(arr, low, high);
      quicksort(arr, low, mid -1);
      quicksort(arr, mid + 1, high);
    }
}
```

What modification you will suggest to reduce the space complexity of the above mentioned tail recursive quicksort() function?

(c) Explain your answer for each of the following questions:

(i) 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?

(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

(ii) Consider the following function to traverse a linked list.

```
void traverse(struct Node *head)
{
  while (head->next != NULL)
  {
    printf("%d ", head->data);
    head = head->next;
  }
}
```

Which of the following is FALSE about the above function?

- The function may crash when the linked list is empty
- The function doesn't print the last node when the linked list is not empty
- The function is implemented incorrectly because it changes head

(iii) You are given the pointer to a node X in a singly linked list. If the pointer, to head node is not given, can you delete the node X from given linked list?

(2 + 1)+ 3 + (2 × 3) =12

5. (a) Explain with an example the reason behind using circular array to implement queue. Why Queue is called an ADT? Write an application of queue.

(b) Write a non-recursive algorithm for generating the elements of the Fibonacci Series.

(3 + 3 + 1) +5 =12

Group – D

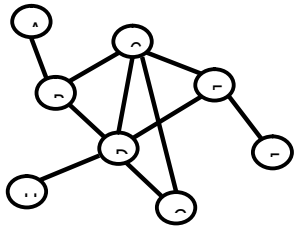
6. (a) Draw the binary tree from given Pre-Order traversal – GBQACKFPDERH and In-Order traversal – QBKCFAGPEDHR.

(b) Given an array **arr[ ]** consisting of the cost of toys and an integer **K** depicting the amount of money available to purchase toys. The task is to find the maximum number of toys one can buy with the amount **K**.
Suggest a suitable data structure to implement it without sorting the data and provide an algorithm to perform the task efficiently.

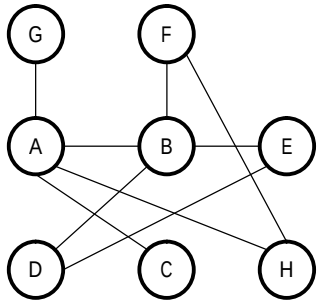
(c) What are the operations that can be done on an AVL tree in sub linear time?

4 + 6 + 2 = 12

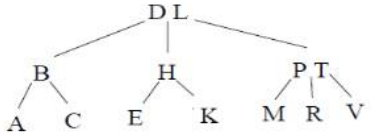
7. (a) Find all the articulation points on the following graph.



(b) Write the pseudo code for BFS traversal and show step wise how you will run your code on the following input graph:



(c) Consider the following B-Tree with a minimum branching factor t = 2.



Show the B-tree that results when deleting P from the given B-Tree.

3 + (3 + 3) + 3=12

Group – E

8. (a) Consider below the pseudo-code that is very similar to Improved Bubble Sort method that was taught in class. There may be one or more mistakes introduced somewhere in this code. Can you identify in which lines the mistakes are? Also suggest the necessary correction so that it performs the intended optimization. You need not copy the whole code; just write the corrected portion along with line number.

```
1.  Bound← n
2.  do
3.  {
4.          t ← 0
5.          for j = 1 to BOUND -1
6.              if key(j) > key(j + 1)
7.                  interchange record(j) ↔ record(j + 1)
8.                      end if
9.              t← j
10.         endfor
11.         BOUND← t
12. } while ( t >= 0)
```

(b) State its best-case and worst-case time complexity of bubble Sort.

(c) Suggest an algorithm to find whether an array is a subset of another array. Note that these arrays contain distinct values.
Example: Input*: arr1[] = {11, 1, 13, 21, 3, 7}, arr2[] = {11, 3, 7, 1}*
Output*: arr2[] is a subset of arr1[]*

5 + 2 + 5=12

9. (a) Consider the following sequence of values: 10, 45, 20, 14, 57, 100, 200, 40, 35, 66, 27, 85, 52, 70 and 18. Suppose we have an array of size 20 and a simple hash function (key % 10) is used to store these values. Two different hashing techniques are used:
i) Separate chaining used
ii) Open addressing
State which technique may suffer from collision and why? Then show how you will tackle these collisions.

(b) Suppose an element occurs twice in an array, show that the expected number of comparisons needed to search the element will be (n + 1) / 3, where n is the total number of elements present in the array.