Question Paper

Exam Date & Time: 31-Dec-2019 (09:30 AM - 12:30 PM)

MANIPAL ACADEMY OF HIGHER EDUCATION

INTERNATIONAL CENTRE FOR APPLIED SCIENCES
END SEMESTER THEORY EXAMINATIONS NOVEMBER 2019
III SEMESTER B.sc. (Applied Sciences) in Engg.
DATA STRUCTURES [ICS 231]

Marks: 100 **Duration: 180 mins.**

Answer 5 out of 8 questions.

1) A) An algorithm that searches a[0:n-1] for the first occurrence of x. The number of comparisons between x and the elements of a isn't uniquely determined by the problem size n. For example, if n = 100 and x = a[0], then only 1 comparison is made. However, if x isn't equal to any of the a[i]'s, then 100 comparisons are made. (12)

```
int sequentialSearch(int a[ ], int n, int x)
{
// search a[0:n-1] for x
int i;
for (i = 0; i < n && x != a[i]; i++);
if (i == n) return -1; // not found
else return i;
}
```

Determine the time complexity for best-case, worst-case and when x = a[j] i.e. express in theta notation of above algorithm using step count method.

B) Write an iterative function to compute a binomial coefficient; then transform it into an equivalent recursive function. Use the following function declaration for both iterative and recursive. (8)

```
int binomialCoeff(int n, int m)
```

2) A) Consider the following stack operations shown in Fig. 1. Write a C++ program [using class/function template] to perform the stack operations. Write the main function to demonstrate the stack operations as shown in Fig.1. (8)

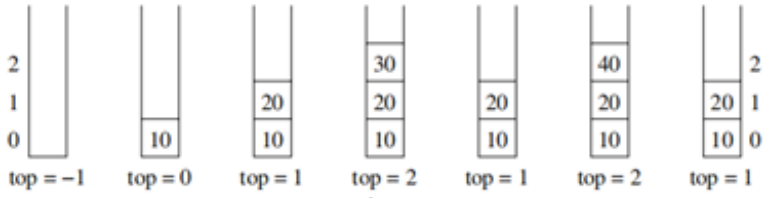

Fig.1 Stack operations

B) Write a C++ program [using class/function template] that implements the infix to postfix notation. The program should read an infix string consisting of single alphabetic characters for variables, parentheses, and the +, -, * (12)

and / operators; call convertion algorithm; and then print the resulting postfix expression. After transforming an algorithm, it should loop and convert another infix string. To test your program, transform the following expressions with your program.
A) (A+B)*C-D*F+C B) (A-2*(B+C)-D*E)*F

3) Consider the following Circular Queue operations shown in Fig. 2. Write a C++ program [using class/function template] to perform the Circular Queue operations. Write the main function to demonstrate the Circular Queue operations as shown in Fig.2. (8)

A)

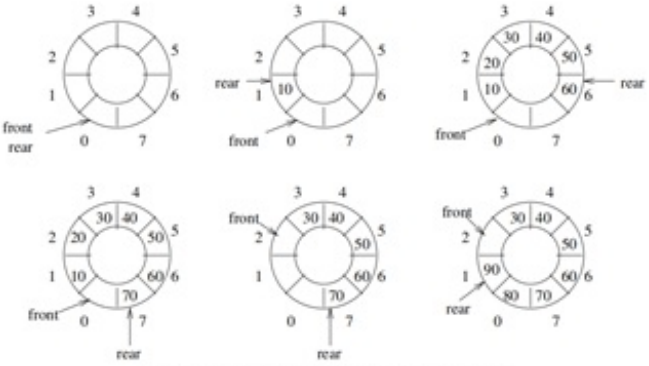

Fig. 2. Circular Queue Operations

B) What are the advantages and disadvantages of linked lists over arrays. (4)

C) Let x =(x1, x2, ..., xn) and y=(y1, y2,.....,yn) be the two singly linked lists. Assume that in each list, the nodes are in non-decreasing order of their data field values. Write a C++ function [use class templates] to merge the two lists to obtain a new z in which the nodes are also in this order. Following the merge, x and y should represent empty lists because each node in x or y is now in z. Use the following method/function declarations only. Also show how it is called in main( ). (8)
Merge(List listOne, List listTwo, List& merged)

4) What are the merits of doubly linked list over singly linked list ? Explain with an example. (4)

A)

B) Write the member functions in C++ [use templates] for doubly linked list, DLinkedList class to insert an item at front, to insert an item at the rear and to delete a given item. Use the following declarations only. The functions should print appropriate message(s). (8)
a)template < class T>
void DLinkedList < T>::InsertFront(T x)
b)template < class T>
void DLinkedList < T>::InsertRear(T x)
c)template < class T>
void DLinkedList < T>::Delete_In-between(T x)

C) In some applications the elements to be put on a queue are already in nodes of type chainNode. For these applications it is desirable to have the methods pushNode(chainNode theNode), which adds theNode at the back of the queue, and popNode, which reomoves and returns the front node of (8)

the queue. Write a C++ code [use template] for these methods.
Use the following methods declarations only.
a) template< class T>
void linkedQueue< T>::pushNode(const T& theElement)
b) template< class T>
void linkedQueue< T> :: popNode()

5) Explain with an example for each, the different storage representations for a binary tree. (4)

A)

B) Suppose you have a binary tree whose data fields are single characters. When the data fields of the nodes are output in inorder, the output is ABCDEFGHIJ, and when they are output in preorder, the output is BAHCEDGFJI. Draw the binary tree showing the data in each node and the pointers between nodes. Show the steps used to arrive at the result. (4)

C) Explain the AVL Tree with balance factor and its all types of rotations for balancing the
tree. Start with an empty AVL tree and perform the following sequence of insertions: 40, 30, 20, 60, 50, 80, 15, 28, 25. Draw the AVL tree for above sequence each insertions and state the rotation type (if any) for each insert. (12)

6) Assume that we have n>=1 distinct integers that are already sorted and stored in the array a[0], ... a[n-1]. Develop a method in C++ is to determine (6)

A) if the integer x is present while considering the conditions such as x< a[mid], x=a[mid], and x>a[mid]. If x=a[j] return j otherwise return -1. And chose the proper name for developed method.
Consider an array contains the elements such as (4,7,8,10,14,21,22,36,62,77,81,91). Using the above developed method, trace the steps followed to find 22. At each loop iteration, including the last, show the contents of first, last and mid.

B) Write all the possible cases, when we delete a node from a binary search tree. Construct the Binary Search Tree (BST) for the list of elements : 10, 5, 20, 14, 30, 8, 6, 9, 35, 25, 12, 17.Considering 10 as the root. Show the BST after every insertion. Remove the nodes 35, 30, 20 and 10 from the above constructed BST and show the BST after every deletion. (14)

7) In Quick Sort, we select a pivot record from among the records to be sorted. (8)

A) Next, the records to be sorted are reordered so that the keys of records to the left of the pivot are less than or equal to that of that of the pivot and those of the right of the pivot are greater than or equal to that of the pivot. Finally, the records to the left of the pivot and those to its right are sorted independently (using Quick Sort method recursively)
Develop a recursive method/function QuickSort, to sort a[1:n], the function invocation is QuickSort(a, 1, n). Function QuckSort assumes that a[n+1] has been set to have a key at least as large as the remaining keys (Use the following declaration).
template < class T>
void QuickSort(T *a, const int left, const int right)
Consider a list of 10 records with keys(26, 5, 37, 1, 61, 11, 59, 15, 48, 19) to sort non-decreasing order. Give the status of the list at each call of

QuickSort

B) In heap sort, the n records are first inserted into an initially empty max heap. Next, the records are extracted from the max heap one at a time. It is possible to create the max heap of n records faster than by inserting the records one by one into an empty heap. For this, write a method/function Adjust in C++, which starts with a binary tree whose left and right subtrees are max heaps and rearranges records so that the entire binary tree is a max heap. The binary tree is embedded within an array using standard mapping. To sort the list, write a method/function HeapSort in C++. Use the following method/function declarations only. (12)

```
template< class T>
void Adjust(T *a, const int root, const int n)
template< class T>
void HeapSort(T *a, const int n)
```

The input list is (82, 90, 10, 12, 15,77, 55, 23). Apply the HeapSort, to sort the list.

8) A) A Graph can be used to show relationships. For example, from the following list of students belonging to the ICAS (Vertices) and their friendships (edges). (8)

Students = {Tanmay, Pritesh, Shubham, Lakshya, Pranoy, Pranav, Avijit}
Friendship={(Tanmay, Shubham), (Lakshya, Pranoy), (Tanmay, Pranav), (Pritesh, Pranoy),
(Pritesh, Lakshya), (Pritesh, Avijit), (Avijit, Lakshya), (Tanmay,Pritesh), (Shubham, Pritesh), (Pranav,Shubham), (Shubham,Pranoy), (Pranav,Pranoy)}

a) Give the adjacency list representation for above complete graph example.
b) Give the Depth-first traversal and Breadth- first traversal for the above complete graph example starting from vertex Tanmay. List the vertices in the order they would be visited.

B) Write the member functions in C++ [use class templates] for All-pairs shortest paths. Use the following declarations only. (8)

```
template< class T>
void MatrixWDigraph::AllLengths(const int n)
```

Use the above member function, compute all pair shortest path for following diagraph shown in Fig. 3.

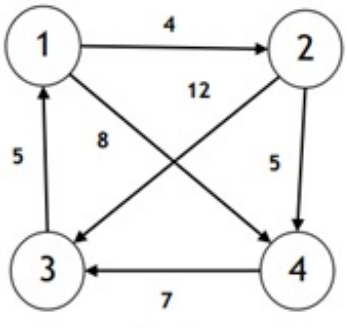

Fig. 3

C) Write a short note on hashing. (4)

-----End-----