PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

END SEMESTER ASSESSMENT (ESA) - Dec 2024

UE23CS252A - Data Structures and its Applications

Total Marks : 100.0

1.a. Given two singly linked lists. The task is to determine if the two linked lists are identical or not. Two linked lists are considered identical if they contain the same data in the same order. If the linked lists are identical, return true otherwise, return false.
Node structure definition is as follows:

```
struct Node {
   int data;
   struct Node* next;
};
```

```
int Identical(struct Node* head1, struct Node* head2) ;
```

Wherein head1 and head2 are the pointers to the singly linked list1 and list2 respectively

Input: List1: 1->2->3->4->5->6, List2: 99->59->42->20
Output: 0
Input: List1: 1->2->3->4->5, List2: 1->2->3->4->5
Output: 1 (5.0 Marks)

1.b. Write the postfix from the following expression using stack:
(i) (p+q)*(m-n)
(ii) a+b*(c^d-e)^(f+g*h)-i (7.0 Marks)

1.c. Write a C function called postfixEval() to evaluate a postfix expression using stack and apply the same for the given postfix expression 6 2 3 + - 3 8 2 / + *
Stack structure definition is as follows:

```
typedef struct stack
{
   int s[MAX];
   int top;
}STACK;
int postfixEval(char postfix[])// returns evaluated value of postfix
expression
// you can use a stack with the standard stack operations. DO NOT write
the push and pop stack methods.
void push(STACK *ps,int ele)
int pop(STACK *ps)
```

(8.0 Marks)

1.d. Write C function for the concatenation of two doubly linked lists
Node structure definition is as follows:

```
struct Node {
   int data;
   struct Node* next,*prev;
};
typedef struct list{
struct Node *head;
}List;

void concatenate(List* p1, List* p2) ;
```

(5.0 Marks)

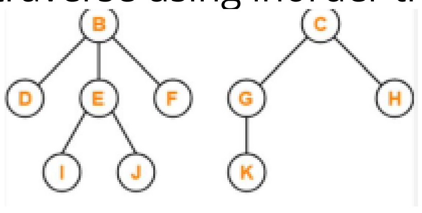
2.a. With suitable example, define:

1. Binary Tree (ii) Level of Binary tree (iii) Complete Binary Tree (iv) Degree of the tree

(8.0 Marks)

2.b. Define Forest. Transform the given forest into a Binary tree and traverse using inorder traversal.

(6.0 Marks)



2.c. What is a Priority queue? Write a function in C to insert an element into a Max Priority queue. Consider an array implementation where the elements are stored in ascending order.
Node structure definition is as follows:

```
typedef struct queue
{
        int info;
        int priority;
}QUEUE;
void enqueue(QUEUE pq[],int data,int priority,int *count)//count is a pointer to number of elements present in the array.
```

(5.0 Marks)

2.d. Write a C function to insert an element into a DEQUEUE from the rear end and to delete an element from the rear end of the DEQUEUE, using linked list implementation.
Node structure definition is as follows:

```
typedef struct node
{
   int data;
   struct node *next;
   struct node *prev;

}Node;
typedef struct dq
{
   Node *front;
   Node *rear;
}Dq;
void insertrear(Dq *obj,int n);// where n is an integer number to be inserted from rear end
void delrear(Dq *obj);
```

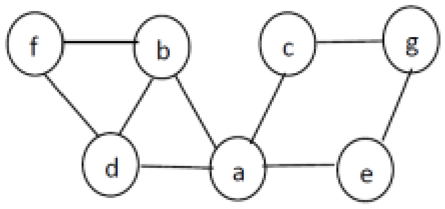
(6.0 Marks)

3.a. Write a C function to traverse a graph using Depth First Search (DFS).
Node structure definition is as follows:

```
struct node
{
        int data;
        struct node *next;
};
void dfs(struct node *a[10],int v,int visited[10]);// where a[10] is the
```

adjacency list ,v is the starting vertex and visited[ ]is the visited array indicating vertex is visited or not
Apply DFS for the graph given below. Consider adjacency matrix implementation(Lexicographic order) and starting vertex as 'a'

(8.0 Marks)



3.b. Define Binary Search tree. Construct a binary search tree (BST) for the following elements: 100, 85, 45, 55, 120, 20, 70, 90, 115, 65, 130, 145.
i. Redraw after 130 is removed from BST
ii.Redraw after 100 is removed from BST(by replacing it with its inorder predecessor) (5.0 Marks)

3.c. What is a heap? Give any three properties of heaps? (5.0 Marks)

3.d. Insert the following elements into an initially empty AVL tree. Show the rotations and all the in between trees.
50 , 20 , 60 , 10 , 8 , 15 , 32 , 46 , 11 , 48 (7.0 Marks)

4.a. What is collision? List the different collision resolution techniques. Explain linear probing with an example. (7.0 Marks)

4.b. Construct a suffix trie for the word banana and show its suffix tree. (5.0 Marks)

4.c. List any four applications of a Trie. Write a function to insert words of lower case in a trie. Also include a function to allocate memory for a Trienode.
Use the following structure of the Trie node.

```
struct trienode {
struct trienode* child[26];
int endofword;
};
```

function prototype:

```
void insert(struct trienode *root, char *key); //root is a pointer to the root and key is word to be inserted
struct trienode *getnode() // allocates memory for a trie node
```

(7.0 Marks)

4.d. You are given a hash table of size 10 (m=10), with the following two hash functions:
$h_1(k)=(2k+3)\%m$
$h_2(k)=(3k+1)\%m$
Insert the keys 3,2,9,6,11,13,7,12 into the hash table using double hashing. (6.0 Marks)