INDIAN STATISTICAL INSTITUTE

End Semestral Examination

M. Tech (CS), 2025-2026 (Semester – I)

Data Structures

Date: 26.11.2025 Maximum Marks: 100 Duration: 3.5 Hours

Note: Answer all questions from Group-A. From Group-B, you can answer as many questions as you like, but the maximum you can score is 40.

You should try to write a proof sketch for the correctness of your algorithms.

Group A

(QA1) Consider the following code snippet for a function `WhatDoYouDo` that is part of a larger program. The program deals with a tree $\mathcal{L}$ whose root node is pointed to by the pointer `root`.

Find out what does the function `WhatDoYouDo` do to $\mathcal{L}$? Explain your answer with suitable diagrams, if needed.

```
struct Node {
    int data;
    struct Node *left, *right;
};

int WhatDoYouDo(struct Node* root) {
    if (root == NULL)   return -1;

    int aa  = WhatDoYouDo(root->left);
    int bb  = WhatDoYouDo(root->right);

    if(aa > bb) return aa+1;
    else return bb+1;
}

int main() {
    /* Create the tree */
    printf("Print the data = %d\n", WhatDoYouDo(root));
    return 0;
}
```

[8]

(QA2) (i) Show that in any binary tree with n nodes, the total number of NULL child pointers, i.e., pointers that do not point to a node is $n + 1$.

(ii) Deduce the maximum number of nodes in a binary tree of height h. Prove your result.

(iii) A *full node* is a node with two children. Prove that the number of full nodes plus one is equal to the number of leaves in a nonempty binary tree.

[4+4+4=12]

(QA3) (i) Define the balance condition for AVL tree.

(ii) Show that an AVL tree with n nodes has height $O(\log n)$.

(iii) Consider inserting a node w in a subtree T_1 of an AVL tree due to which a node z has become unbalanced. Let z also be the lowest ancestor of the subtree T_1 where imbalance has happened due to the insertion of w in T_1. Let y be the left child of z and x be the left child of y. Let subtree T_1 and T_2 be the left and right child of x, respectively. Let T_3 be the right subtree of y and T_4 be the right subtree of z. Let height of T_1 was originally h and due to insertion of w, it has become $h + 1$.

- Describe a method to show how a LL (single) rotation restores balance at z. What is the time complexity of your method to restore balance?
- Prove that your described method does not need to restore balance for any node in the path from z onwards to the root of the tree.

[2+6+(8+9)=25]

(QA4) (i) Describe the random skip list data structure.

(ii) Show that the expected space requirement of a random skip list for a set S of size n is $O(n)$.

(iii) Show that the number of levels r in a random leveling of a set S of size n, has $r = O(\log n)$ with high probability.

[7+4+4=15]

Group B

(QB1) Show how to implement the stack ADT using only a priority queue and one additional member variable. [10]

(QB2) You are given an array of distinct integers which was initially sorted in ascending order, but then rotated at an unknown pivot. As an example, $[4, 7, 8, 9, 0, 1, 2]$ is a rotation of $[0, 1, 2, 4, 7, 8, 9]$. You are also given an integer X. Design an efficient algorithm to determine if X exists in the array. [10]

(QB3) Consider a singly linked list $\mathcal{L}$ of n nodes containing the elements in the following sequence:

$$L_1 \to L_2 \to L_3 \to \cdots \to L_{n-2} \to L_{n-1} \to L_n$$

We need to reorder L to the following:

$$L_1 \to L_n \to L_2 \to L_{n-1} \to L_3 \to L_{n-2} \to \cdots.$$

Design an in-place algorithm that without changing node values and using only pointer manipulation does the reordering. Your algorithm can use only $O(1)$ extra space. [10]

(QB4) Consider a depth first search on an undirected graph G. We maintain a clock on the event when the search visits a node.

(i) For each node $v \in G$, let pre[v] denote the time when the search discovers node v for the first time and post[v] denote the final departure time of the search from node v. Show that for any two nodes u and v, the two intervals [pre[u], post[u]] and [pre[v], post[v]] are either disjoint or one is contained within the other.

(ii) Design and analyze an efficient method to find out whether G is acyclic.

[4+6=10]

(QB5) (i) Consider a one-dimensional range search, with the interval $[x : x']$ on a set of n real numbers X stored in a balanced binary search tree T. A *canonical subset* of v is the subset of points stored in the leaves of the subtree rooted at a node v. For example, the canonical subset corresponding to the root of the tree T, is the whole point set P.

Deduce an asymptotic bound on the number of canonical subsets that are encountered by the one-dimensional range search on T.

(ii) Deduce and explain the recurrence for a query with an axis-parallel rectangle in a kd-tree storing n points in 2D. You need not solve the recurrence.

(iii) Now consider the same kind of query with an axis-parallel hyper-rectangle in a kd-tree storing n points in dimension d. Show how you can generalize the above search to dimension d. Deduce and explain the recurrence formed. You need not solve the recurrence.

[2+4+4=10]

(QB6) (i) Define a universal family of hash functions.

(ii) Describe the chaining method for resolving *collisions* in a hash table.

(iii) Find out the expected length of the chain corresponding to each entry of the hash table when the hash function is chosen uniformly at random from a universal family of hash functions.

[1+3+6=10]