

# COL106 Minor 2

ARPIT SAXENA

TOTAL POINTS

42 / 70

## QUESTION 1

Q1 11 pts

1.1 Q1-a 8 / 8

- ! + 1 pts Correct Insertion/Probes for 873/837: 3
- ! + 1 pts Correct Insertion/Probes for 9734/8843: 3, 1
- ! + 1 pts Correct Insertion/Probes for 280/640 : 0
- ! + 1 pts Correct Probes for deleting 9734/8843 : 3, 1
- ! + 2 pts Probes for searching 143/323 : 3, 1, 4
- ! + 2 pts Correct Insertions/Probes for inserting 14/32: 0, 3, 1
- ff - 1.5 pts Did not listed probes
- ff + 0 pts Incorrect/Unattempted

1.2 Q1-b 1.5 / 3

- ! + 1.5 pts Null slot found then stop search
- ff + 1.5 pts Probed the slots in correct order until repeated slot is found (when table is full and element not present) or element found
- ff + 0 pts Incorrect/unattempted

## QUESTION 2

2 2 5 / 5

- ! + 2 pts Main final structure
- ff - 0.5 pts Not Setting parents field of p and g
- ! + 2 pts Correctly attaching old children of n
- ff - 0.5 pts Not setting parent field of "Old Children of n"
- ! + 1 pts Correctly setting new parent of n
- ff + 0 pts Incorrect Solution

## QUESTION 3

3 10 pts

3.1 Q3-Delete 0 / 6

- ! + 0 pts Incorrect/Unattempted

ff + 1 pts Correctly handles no children case

ff + 1 pts Correctly handles the case when the key values matched, left subtree is NULL and right subtree is not NULL

ff + 2 pts Correctly handles the case when the key values matched, left subtree is not NULL and right subtree is NULL ; also include recursing again on left subtree

ff + 2 pts Correctly handles 2 children case : Includes finding the inorder successor/predecessor and deleting that predecessor/successor and recursed again on root->left

ff + 1 pts Incorrect handling of 2 children case

ff + 1 pts Only deleted the first key found

ff - 0.5 pts Not mentioned about how to delete the successor or predecessor or incorrect successor deletion

ff - 1 pts Didn't recurse on root->left after deleting

ff + 2 pts Incomplete/Partial Correct/ Specific Case only

3.2 Q3-Search 4 / 4

ff + 0 pts Incorrect/Unattempted

! + 1 pts Correct base case i.e. termination condition

! + 0.5 pts Implemented only the Binary Search

! + 1 pts Correctly search duplicates keys

! + 1 pts return the duplicates keys

! + 0.5 pts pseudo code attempt

ff + 0 pts Wrong search

storing is not allowed.

## QUESTION 4

4 Q4 2 / 5

! + 2 pts - Used global lock/ synchronized

- Mentioned about locking the root

ff + 1 pts - Incorrect usage of per node lock - leading to inconsistent results.  
- Incomplete answer regarding lock/ synchronized.  
But in the right direction.  
ff + 0 pts No explanation of how to implement synchronization or entirely incorrect. Not mentioning anything about how to synchronize.  
Not answered.  
ff + 3 pts Justification - the deadlock issue if the root is not locked

#### QUESTION 5

5 Q5 3 / 5

ff + 0 pts Incorrect/Unattempted  
ff + 1 pts lock whole list without proper explanation  
ff + 1.5 pts lock whole list with proper explanation  
! + 1 pts Correct and precise synchronization for search with incorrect explanation  
ff + 2 pts Correct and precise synchronization for search with correct explanation  
! + 1 pts Correct and precise synchronization for insert with incorrect explanation  
ff + 1.5 pts Correct and precise synchronization for insert with correct explanation  
! + 1 pts Correct and precise synchronization for delete with incorrect explanation  
ff + 1.5 pts Correct and precise synchronization for delete with correct explanation

#### QUESTION 6

6 Q6 7 / 10

ff + 0 pts Incorrect or Unattempted  
ff - 0.5 pts Didn't observe that all leaves have the same height  
! + 1.5 pts Part A: Observed that minimum height will happen when each node is 4-node.  
! + 2 pts Part A: Derived Correct expression for maximum number of keys for a given height  
! + 0.5 pts Part A: Derived Correct expression for minimum height given number of keys n  
! + 1.5 pts Part B: Observed that maximum height will happen when each node is 2-node.

! + 1.5 pts Part B: Derived correct expression for maximum height given number of keys n

ff + 1.5 pts Part C: Observed that to find a key, number of nodes examined is at most the maximum height

ff + 1.5 pts Part C: Observed that maximum number of comparisons per node are 2 (for 3-nodes and 4-nodes).

ff - 1 pts PartC: not mentioned why 2 comparison in part c

ff + 0 pts Didn't notice that n denotes number of keys, not number of nodes

ff - 1 pts PartC: maximum height not mentioned

#### QUESTION 7

7 Q7 6 / 6

! + 6 pts Correct

ff + 2 pts If algorithm partially or fully correct but not  $O(h)$

ff + 3 pts If algorithm not written but worked out by example

ff + 0 pts If algorithm not written or incorrect

ff + 1 pts Partial marks (Brute-force or almost brute force kind of approach or just writing select root as the one of the roots and attach remaining as subtree). A recursive approach on nodes is brute-force.

ff + 0 pts Click here to replace this description.

#### QUESTION 8

8 Q8 0.5 / 6

ff + 3 pts correct explanation

ff + 3 pts correct answer

ff + 0 pts Incorrect or blank

! + 0.5 pts explanation provided but not appropriate

#### QUESTION 9

9 Q9 0 / 6

! + 0 pts Incorrect/Didn't attempt

ff + 1 pts Recognition that Preorder traversal breaks into Root followed by left subtree and right subtree

ff + 2 pts Recognition that elements in the left subtree are smaller than the root and, scanning to get there

££ + 2 pts Proper book keeping in the pseudocode

££ + 1 pts Proper justification

#### QUESTION 10

10 Q10 5 / 6

! + 4 pts Correct non recursive algorithm

! + 1 pts Time complexity  $O(n)$  provided algorithm is correct

££ + 1 pts Argument for correctness

££ + 1 pts Correct recursive solution

££ + 0 pts Incorrect/ Unattempted

££ + 0 pts Does not find depth of binary tree, stops after finding depth of first leaf













