<u>CSCE 500</u> Midterm Exam

1. For any n-key <u>B-tree of height h</u> and with the minimum node degree of $t \geq 2$, <u>prove that</u> h is no larger than $\log_t \frac{n+1}{2}$. (Hint: consider the number of keys stored in each tree level.)

2. The utilization efficiency of a hash table depends heavily on its hashing function(s) employed. Explain briefly (1) how <u>perfect hashing</u> works, and (2) how <u>Cuckoo hashing</u> works under two hash functions of h_1 and h_2.

3. The <u>binary search tree</u> (T) facilitates key search and it involves several operations to maintain the tree property when a node (z) is deleted, as shown in the following pseudo code, TREE-DELETE(T, z), where TRANSPLANT(T, u, v) replaces the subtree rooted at u with one rooted at v. <u>Fill</u> in those <u>three missing statements</u> in the pseudo code below and <u>sketch</u> an example binary search tree to illustrate such a deletion case.

```
TREE-DELETE(T, z)
if z.left == NIL
    TRANSPLANT(T, z, z.right)          // z has no left child
elseif z.right == NIL
    TRANSPLANT(T, z, z.left)           // z has just a left child
else // z has two children.
    y = TREE-MINIMUM(z.right)          // y is z's successor
    if y.p ≠ z
        // y lies within z's right subtree but is not the root of this subtree.



    // Replace z by y.
    TRANSPLANT(T, z, y)
    y.left = z.left
    y.left.p = y
```

4. Given the initial <u>B-tree</u> with the minimum node degree of <u>$t = 3$</u> below, show the results (a) <u>after deleting</u> two keys in order: M then R and (b) followed by <u>inserting</u> the key of L_l, with $L < L_l < L_2$.