

# CS 32

## Exam 3 - Answer Key

April 24, 2015

### General Instructions

- Answer the items completely. Show your solutions/justifications when asked.
- Write as legibly as possible. Illegible or unreadable answers and solutions may not merit any points.
- Refrain from making unnecessary motions and sounds during the exam. Any suspicious behavior will be dealt with accordingly.
- Direct all questions to the proctor.
- If you need to go to the CR, hand your questionnaire, answer sheet, and scratch paper to the proctor before heading out. Only one person at any given time is allowed to go out.
- Once you're done with the exam (one way or the other), submit your scratch papers and the questionnaire together with your answer sheet.

### Questions

Consider the following generalized list  $L = (((1), (2, 3)), ( ), (4, (5, 6), 7, (8, (9, 10))))$

1. Assume that the only allowable functions to operate on  $L$  are *head* and *tail*. Answer the following questions (*0.2 point each*):
  - (a) How do you access the atomic element 3?  
**ANSWER:** `head(tail(head(tail(head(L)))))`
  - (b) How do you access the null list element inside  $L$ ?  
**ANSWER:** `head(tail(L))`
  - (c) How do you access the list element (9,10) inside  $L$ ?  
**ANSWER:** `head(tail(head(tail(tail(tail(head(tail(tail(L))))))))))`
  - (d) What will the expression `head(tail(head(tail(tail(L)))))` evaluate to?  
**ANSWER:** (5,6)
  - (e) What will the expression `head(tail(tail(head(tail(tail(L))))))` evaluate to?  
**ANSWER:** 7



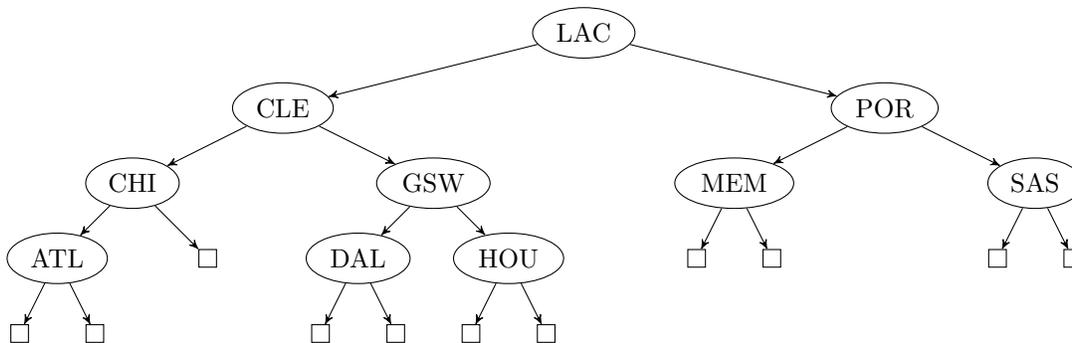
Consider the following keys to be inserted into an AVL tree:  
 GSW, ATL, HOU, LAC, MEM, SAS, CLE, POR, CHI, DAL

3. Construct the AVL tree from the keys above by inserting them in the order that they are listed above (left to right). Show the status of the tree upon insertion of a key, and then after rotations (if any) to maintain AVL property. If rotations are performed, just mention how the rotations and made and on what key.

**ANSWER:** Steps in inserting the keys into the AVL tree are as follows:

- (a) Insert GSW
- (b) Insert ATL as left son of GSW
- (c) Insert HOU as right son of GSW
- (d) Insert LAC as right son of HOU
- (e) Insert MEM as right son of LAC
- (f) Perform left rotation on HOU, making LAC the new right son of GSW, and HOU the left son of LAC
- (g) Insert SAS as right son of MEM
- (h) Perform left rotation on GSW, making LAC the new root of the tree, GSW the left son of LAC, and HOU the right son of GSW
- (i) Insert CLE as right son of ATL
- (j) Insert POR as left son of SAS
- (k) Perform right rotation on SAS, making POR the new right son of MEM, and SAS the right son of POR
- (l) Perform left rotation on MEM, making POR the new right son of LAC, and MEM the left son of POR
- (m) Insert CHI as the left son of CLE
- (n) Perform right rotation on CLE, making CHI the new right son of ATL, and CLE the right son of CHI
- (o) Perform left rotation on ATL, making CHI the new left son of GSW, and ATL the left son of CHI
- (p) Insert DAL as right son of CLE
- (q) Perform left rotation on CHI, making CLE the new left son of GSW, and CHI the left son of CLE
- (r) Perform right rotation on GSW, making CLE the new left son of LAC, GSW the right son of CLE, and DAL the left son of GSW

Resulting AVL tree:

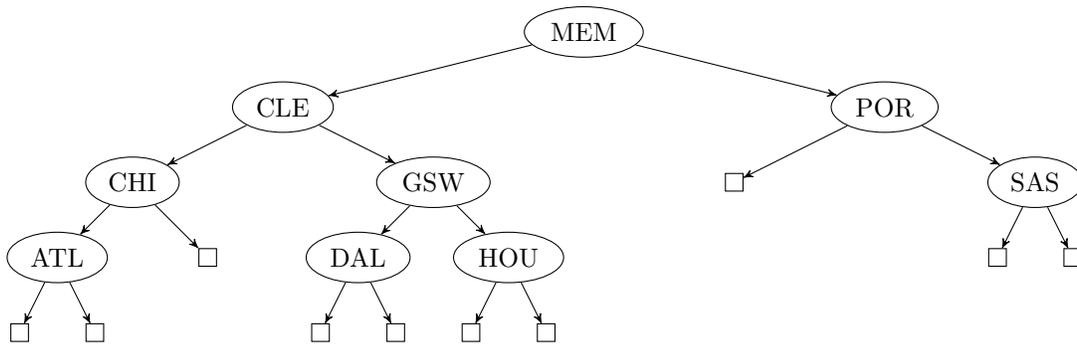


4. Show what happens to the AVL tree upon deletion of the root node. Show the status of the tree after deletion, after node succession, and whenever possible, after rotation/s.

**ANSWER:** Steps in deletion are as follows:

- (a) Delete LAC
- (b) Assign MEM as new root

Resulting AVL tree:



## Scoring Mechanics

1. For Item 1: Each sub-item will be evaluated via an **all-or-nothing** criterion
2. For Item 2: **0.1 point deduction** is given for each erroneous feature.
3. For Item 3: **0.1 point deduction** for each erroneous insertion or balancing was made. **An additional 0.1 point deduction** is given if the final AVL tree is erroneous.
4. For Item 4: **0.2 point deduction** for each mistake in the steps. **An additional 0.1 point deduction** is given if the final AVL tree is erroneous. If the AVL tree created in Item 3 is erroneous, the points garnered in this item is halved.