Practical Task 7.1

(Pass Task)

Submission deadline: 11:59 pm Sunday, May 9th
Discussion deadline: 11:59 pm Sunday, May 30th

Task Objective

In this task you will answer a series of questions regarding the processing of AVL trees. These involve drawing figures neatly to show the process followed.

Background

A Binary Search Tree (BST) is a rooted tree where each node stores a key that is greater than the all the nodes in in its left subtree and less than all the nodes in the right subtree. By organising data in this way, we can instigate fast insertions, lookups and removals of data. This allows for a theoretical possible processing time of $log\ n$ in many cases. However, the tree can become unbalanced which can cause the processing to blow out to a worst case of $O(n)$.

An AVL tree (named after its inventors Adelson-Velsky and Landis) provides a process for efficiently maintaining a balanced binary search tree. The operations required during each insert and removal run in constant time. As the tree is guaranteed to be balanced an AVL tree can ensure a worst time complexity of $O(log\ n)$ for all operations – significantly improving performance.

Formally a binary search tree can be called an AVL-tree if for each $v \in T, b(v) \in \{-1, 0, 1\}$ holds, where $b(v) = h(T_l) - h(T_r)$; $h(T)$ is the height of a tree T; v is a node in a tree; and, T_l and T_r are the left and right subtrees of v. In other words, if the height of the left tree and the right tree only differ by one for every node in the tree then we can the binary search tree is an AVL tree.

To ensure the tree maintains this status there are unique operations used for insertion and deletion of a node. While related to the BST approach insertion involves performing a rebalancing operation after insertion by performing a rebalancing of the tree. The exact nature of this rebalancing is dependent on where the node was added and can involve either performing a left rotation, right rotation, right-left rotation, or a left-right rotation. Similarly, deletion also starts in the same way as for the BST. However, the resulting tree has the risk of being out of balance and so deletion requires a rebalancing step as well.

Task Details

Your task is to answer all the following questions and complement each answer with a detailed explanation.

1. Draw a series of figures demonstrating the insertion of the values.

   17, 13, 1, 8, 5, 11, 24, 27, 15, 6, 16

   into an initially *empty* AVL tree. Insert the values in the order they appear in the given sequence. You must

   - Show the resulting AVL tree immediately before and after each insertion step that causes the tree's rebalancing.
   - Calculate the ***balance degree*** (as the difference between the heights of the left and the right subtrees rooted at a node) and label each node of the AVL tree before and after the necessary rebalancing.
   - Clearly indicate the node(s) at which rotation is performed. Here, let v be the first node you encounter in going up from the newly added node, say z, toward the root of the AVL tree T such that v is unbalanced. And let x denote the child of v with greater height (and note that x must be an ancestor of z). Determine whether the subtree of T rooted at v is *right* or *left heavy*. Similarly, specify whether the subtree of T rooted at x is *right* or *left heavy*.
   - Performing a *single* or a *double rotation* as the rebalancing (repairing) operation on the AVL tree, specify the type of the rotation that you apply: ***Single Left Rotation***, ***Single Right Rotation***, ***Left-Right Rotation***, or ***Right-Left Rotation***.
   - Each time a new value has been inserted, ensure that both the *Binary Search Tree Property* and the *Height-Balance* (*AVL Tree*) *Property* are maintained.

   Note: ensure you practice this process of adding items with different numbers. During the interview you may be asked to add additional nodes and explain the process.

2. Now, draw a series of figures showing the deletion of the values:

   17, 15, 11, 24, 27, 13, 16, 5, 6, 1, 8

   from the AVL tree built in the previous part of the task. Delete the values in the order they appear in the given sequence. Draw the AVL tree after each deletion and rotation (if any), and complement each of the figures with all the aforementioned details.

   When you delete a node with two children, you must always replace it with the ***largest element smaller than the key of the deleted node*** (as opposed to another possible rule: the smallest element larger than the key of the deleted node). **Following this rule is important** to let us check your solution quickly

   Note: ensure you practice this process of deleting items. During the interview you may be asked to delete different nodes and explain the process.

3. What order should we insert the elements $\{1, 2, \dots, 7\}$ into an empty AVL tree so that we do not have to perform any rotations on it?

Further Notes

- The lecture notes of weeks 6 and 7 should be a good starting point outlining the topic of AVL and binary search trees.
- You will learn how to complete this task by reading sections 11.1.1, 11.1.2, and 11.3.1 of the course book “Data Structures and Algorithms in Java” by Michael T. Goodrich, Irvine Roberto Tamassia, and Michael H. Goldwasser (2014). You may access the book on-line for free from the reading list application in CloudDeakin available in Resources → Additional Course Resources → Resources on Algorithms and Data Structures → Course Book: Data structures and algorithms in Java.
- You may find the attached tutorial on the AVL tree rotations useful to clarify how different rotations work and when each of them must be applied for the purpose of rebalancing.

Marking Process and Discussion

To get your task completed, you must finish the following steps strictly on time.

1. Work on your task either during your allocated lab time or during your own study time.
2. Once the task is complete you should make sure that your program implements all the required functionality, is compliable, and has no runtime errors. Programs causing compilation or runtime errors will not be accepted as a solution. You need to test your program thoroughly before submission. Think about potential errors where your program might fail. Note we can sometime use test cases that are different to those provided so verify you have checked it more thoroughly than just using the test program provided.
3. Submit your solution as an answer to the task via the OnTrack submission system. This first submission must be prior to the submission “S” deadline indicated in the unit guide and in OnTrack.
4. If your task has been assessed as requiring a “Redo” or “Resubmit” then you should prepare a new submission. You will have 1 (7 day) calendar week from the day you receive the assessment from the tutor. This usually will mean you should revise the lecture, the readings indicated, and read the unit discussion list for suggestions. After your submission has been corrected and providing it is still before the due deadline you can resubmit.
5. If your task has been assessed as correct, either after step 3 or 4, you can “discuss” with your tutor. This first discussion must occur prior to the discussion “D”.
6. Meet with your tutor or answer question via the intelligent discussion facility to demonstrate/discuss your submission. Be on time with respect to the specified discussion deadline.
7. The tutor will ask you both theoretical and practical questions. Questions are likely to cover lecture notes, so attending (or watching) lectures should help you with this compulsory interview part. The tutor will tick off the task as complete, only if you provide a satisfactory answer to these questions.
8. If you cannot answer the questions satisfactorily your task will remain on discussion and you will need to study the topic during the week and have a second discussion the following week.
9. Please note, due to the number of students and time constraints tutors will only be expected to mark and/or discuss your task twice. After this it will be marked as a “Exceeded Feedback”.
10. If your task has been marked as “Exceeded Feedback” you are eligible to do the redemption quiz for this task. Go to this unit’s site on Deakin Sync and find the redemption quiz associated with this task. You get three tries at this quiz. Ensure you record your attempt.
    I. Login to Zoom and join a meeting by yourself.
    II. Ensure you have both a camera and microphone working
    III. Start a recording.
    IV. Select Share screen then select “Screen”. This will share your whole desktop. Ensure Zoom is including your camera view of you in the corner.
    V. Bring your browser up and do the quiz.
    VI. Once finished select stop recording.

VII. After five to ten minutes you should get an email from Zoom providing you with a link to your video. Using the share link, copy this and paste in your chat for this task in OnTrack for your tutor to verify the recording.

11. Note that we will not check your solution after the submission deadline and will not discuss it after the discussion deadline. If you fail one of the deadlines, you fail the task and this reduces the chance to pass the unit. Unless extended for all students, the deadlines are strict to guarantee smooth and on-time work through the unit.
12. Final note, A “Fail” or “Exceeded Feedback” grade on a task does not mean you have failed the unit. It simply means that you have not demonstrated your understanding of that task through OnTrack. Similarly failing the redemption quiz also does not mean you have failed the unit. You can replace a task with a task from a higher grade.