DS 4300 - Spring 2025

Homework 02

Extra Credit Deadline: Monday, January 27 2024 @ 11:59 pm
Regular Credit Deadline: Wednesday, January 29 2024 @ 11:59 pm
(Note that these are one day off from the typical Sunday/Tuesday submissions)

Directions:
1. From the File Menu, make a copy of this document in your own Google Drive account OR download an MS Word version.
2. Provide **professional-quality, type-set** solutions for each of the stated questions.
   - Do not delete the actual question statement. However, you may add additional space between the questions and extra pages to accommodate your solutions.
   - ***Your solutions must be typeset***. The solutions you incorporate into this document MAY NOT be handwritten. This includes diagrams/trees. A good option is Google Draw which can be inserted directly into the document from the *Insert* menu. This precludes writing them on an iPad-like device and copying them (or a screenshot) into this document.
   - To reiterate, handwritten solutions in any form will receive NO CREDIT.
3. **Generate a PDF of your document with your solutions**. DO NOT upload screenshots, images, or pictures. Reminder, DO NOT delete the question statements. If you compose your solutions in LaTeX, please retype the questions.
4. Upload your solutions document to GradeScope by the due date. **It is your responsibility to associate each question with your solution in GradeScope and click the submit button afterward.** Failure to do so may result in no credit for any questions not linked with your solution *and will not be a valid reason to request a re-grade*.

Academic Collaboration Reminder:
Remember that you may not look at, copy, capture, screenshot, or otherwise take possession of any other students' solutions to any of these questions. Further, you may not provide your solutions in part or in whole to any other student. Doing any of the above constitutes a violation of academic honesty which could result in an F in this class and a referral to OSCCR.

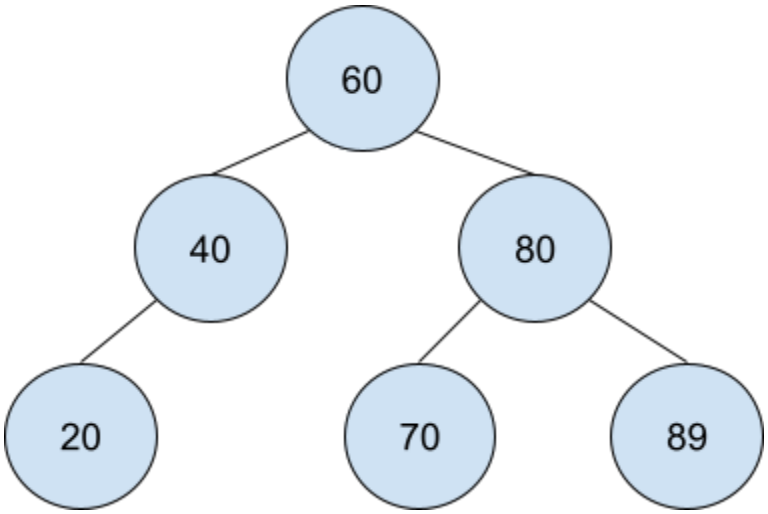
What is permissible? You are free and encouraged to talk to your peers about the conceptual material from the lectures or the conceptual material that is part of this assignment. I'm confident you know where the line between collaborative learning and cheating sits. Please don't cross that line.

Question 1:

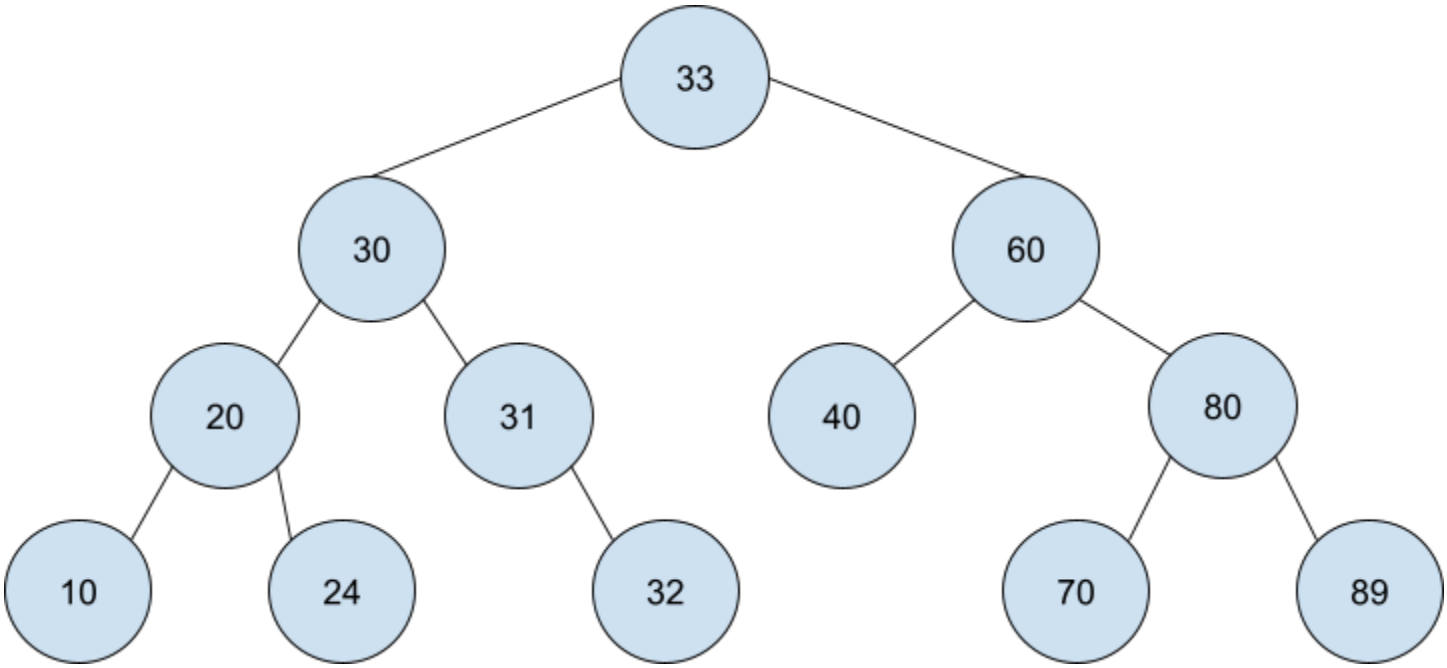
Each of the following key-value pairs in the list below represents a document (the integer) and a word in that document (the letter) from your P01 data set. Insert the key:value pairs sequentially from (20:O) to (32:E) into an initially empty AVL tree. Use the integer as the key to insert. *Note*: this will not produce an inverted index like you're creating for the practical. You only need to provide the balanced state of the tree after inserting (70:E) and (32:E).

[(20:O), (40:S), (60:T), (80:R), (89:N), (70:E), (30:T), (10:N), (33:A), (31:H), (24:R), (32:E)]

Balanced state of the tree after inserting (70:E):



Balanced state of the tree after inserting (32:E):



Provide a list of all insertions that caused an imbalance, at what node the imbalance was found, and what imbalance case was found. You can use the following as a template:
When inserting _____, the tree became imbalanced at _____ and was found to be a case _____. (you can use the number 1, 2, 3, 4 or letter version LL, LR, RL, RR)

When inserting 60, the tree became imbalanced at node 20 and was found to be a case RR.

When inserting 89, the tree became imbalanced at node 60 and was found to be a case RR.

When inserting 70, the tree became imbalanced at node 40 and was found to be a case RL.

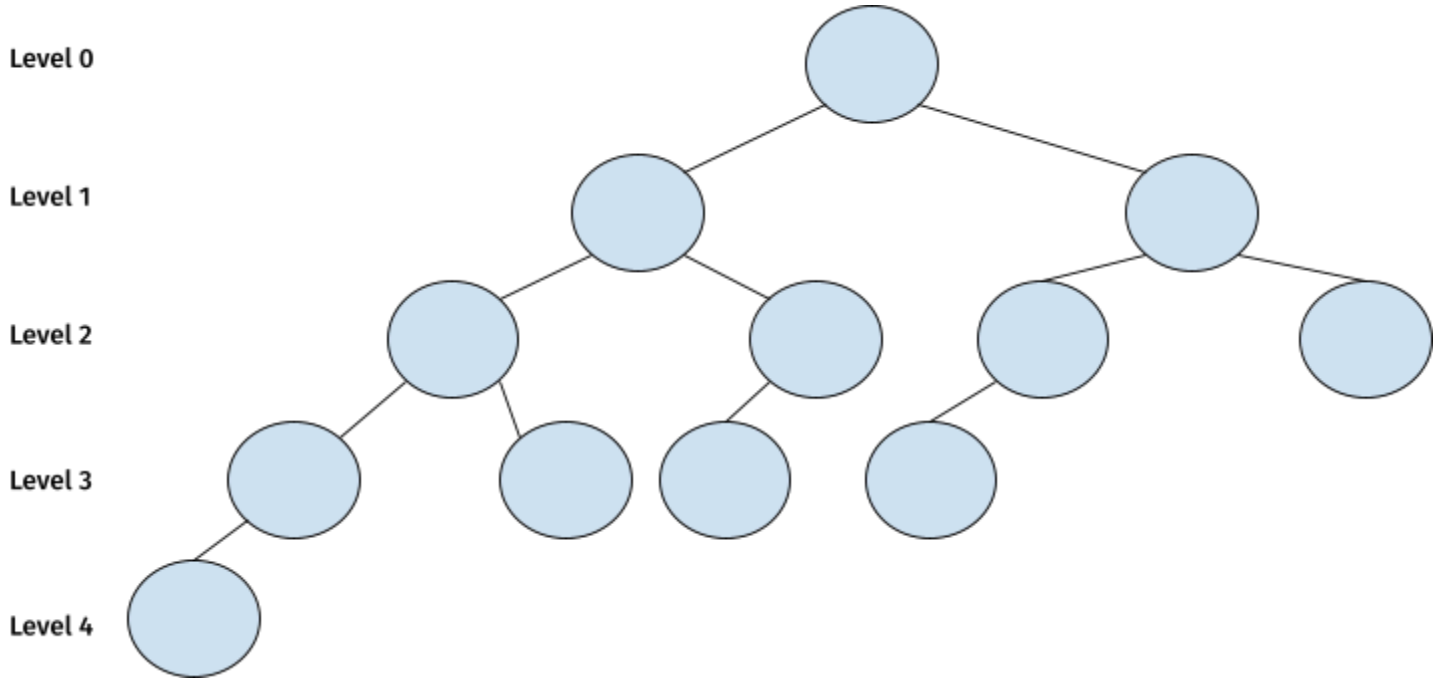
When inserting 30, the tree became imbalanced at node 40 and was found to be a case LR.

When inserting 31, the tree became imbalanced at node 40 and was found to be a case LL.

When inserting 32, the tree became imbalanced at node 60 and was found to be a case LR.

Question 2:

Draw an AVL Tree of height=4 (a tree with 4 levels) with the *minimum* number of nodes possible. You can draw circles to represent the nodes; they do not need to contain values.



Question 3:

Insert the tuples of the form (A:B) from question 1 into a hash table of size 10. The hash table should use separate chaining for collision resolution.

Version A: Use the hash function h(A) = A mod 10.

Index	
0	**(20:O), (40:S), (60:T), (80:R), (70:E), (30:T), (10:N)**
1	**(31:H)**
2	**(32:E)**
3	**(33:A)**
4	**(24:R)**
5	
6	
7	
8	
9	**(89:N)**

Version B: Use the h(B) = lookupASCII(B) mod 10. ASCII is a character encoding that maps a single character to an integer value. For example, the ASCII code of the letter M is 77. You can find an ASCII chart > here <.

Index	
0	
1	
2	**(80:R), (31:H), (24:R)**
3	**(40:S)**
4	**(60:T), (30:T)**
5	**(33:A)**
6	
7	
8	**(89:N), (10:N)**
9	**(20:O), (70:E), (32:E)**

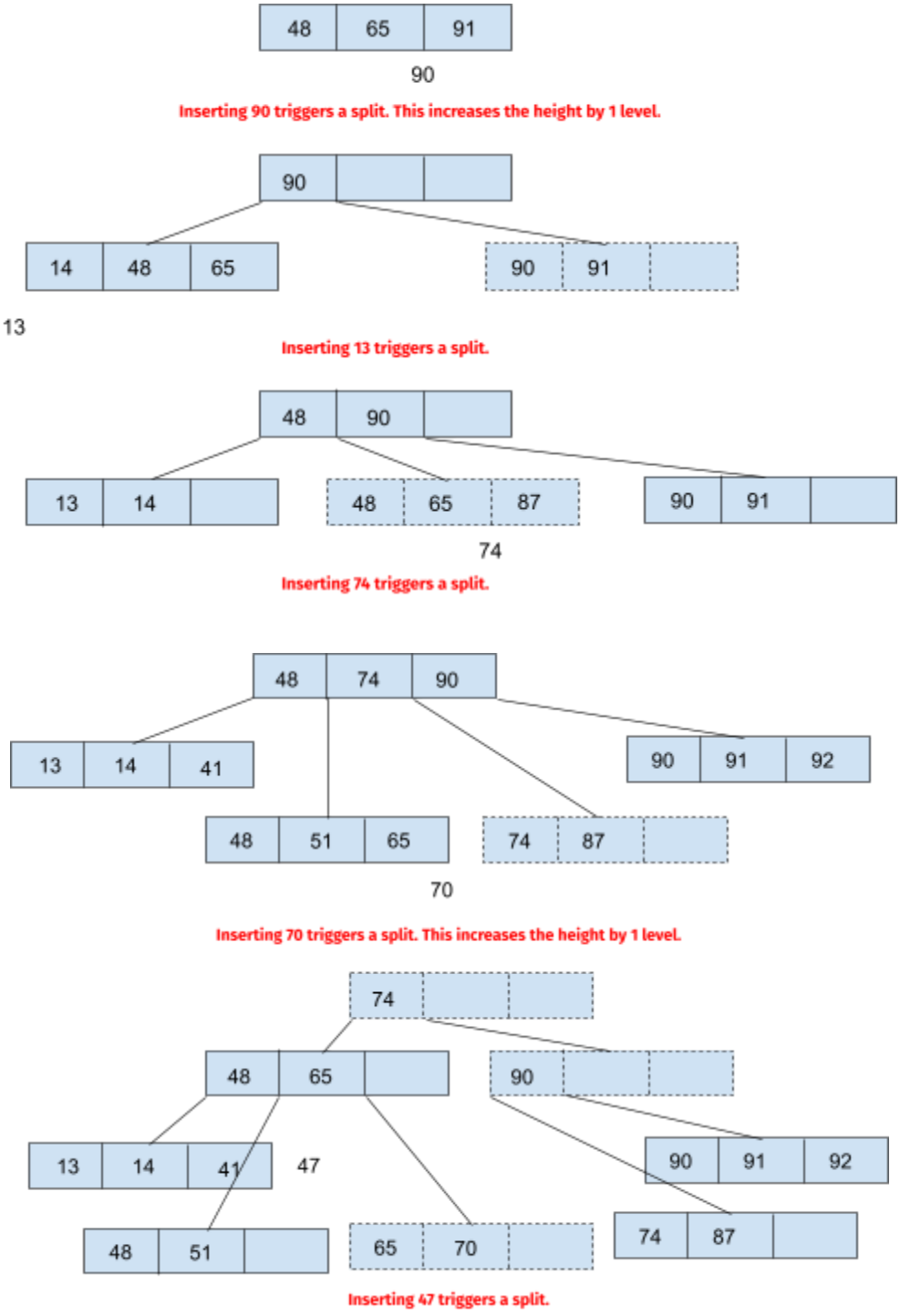
Question 4:

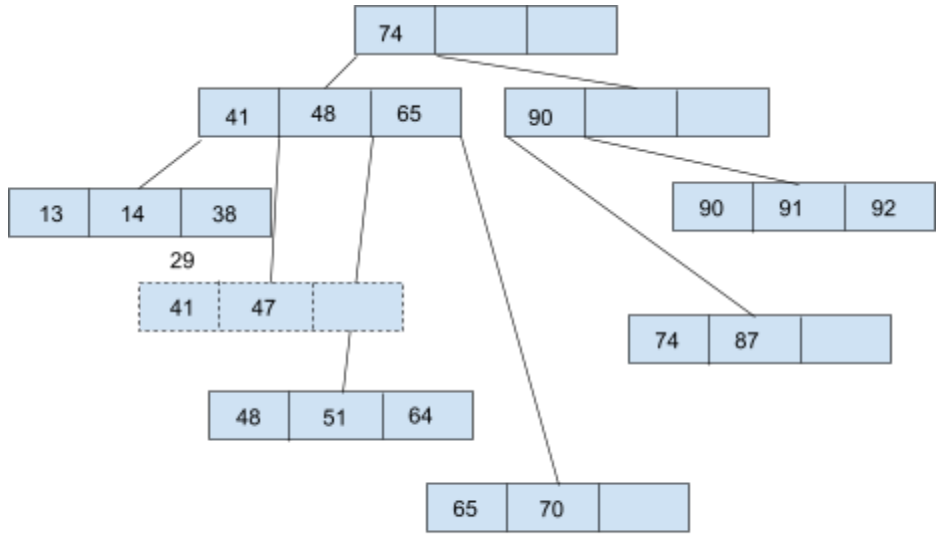
Insert the following list of integers into a B+ Tree where M = 3. That is, each internal node contains max 3 keys and 4 children and each leaf node contains max three keys. When splitting a node due to overflow, leave 1 element in the left node and move 2 elements to the newly created right node.

[48, 65, 91, 90, 14, 13, 87, 74, 51, 92, 41, 70, 47, 64, 38, 29, 50, 21]

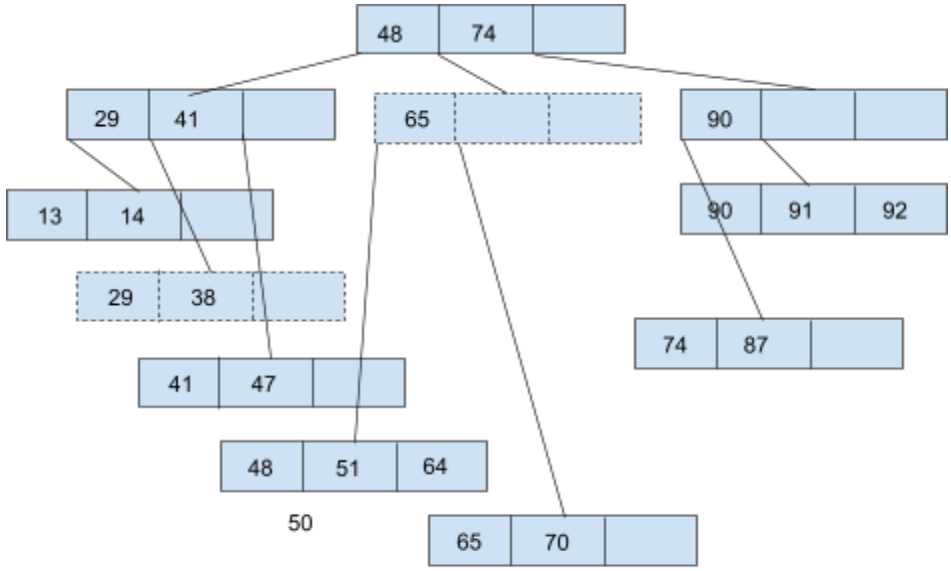
Which insertions triggered a node split? **[90, 13, 74, 70, 47, 29, 50]**
Which insertions increased the height of the tree by 1 level? **[90, 70]**

Step-by-step:





Inserting 29 triggers a split.



Inserting 50 triggers a split.