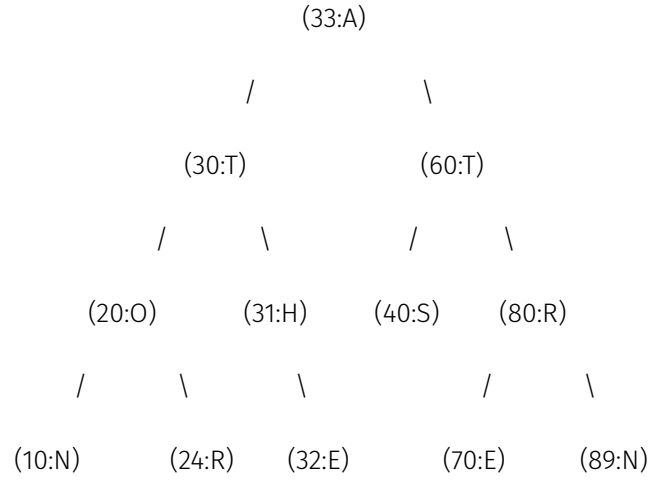
Question 1:

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)]

After inserting (70:E):

After inserting (32:E):

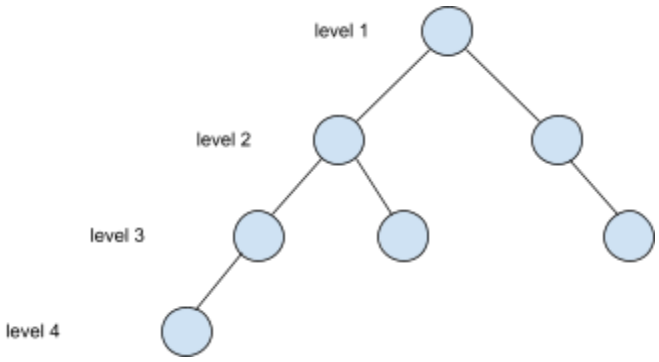


Provide a list of all insertions that caused an imbalance, at what node the imbalance was found, and what imbalance case was found. You can use the following as a template:
When inserting _____, the tree became imbalanced at _____ and was found to be a case _____. (you can use the number 1, 2, 3, 4 or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

When inserting (60:T), the tree became imbalanced at (20:O) and was found to be a case RR.
When inserting (89:N), the tree became imbalanced at (60:T) and was found to be a case RR.
When inserting (70:E), the tree became imbalanced at (40:S) and was found to be a case RL.
When inserting (30:T), the tree became imbalanced at (40:S) and was found to be a case LR.
When inserting (31:H), the tree became imbalanced at (40:S) and was found to be a case LL.
When inserting (32:E), the tree became imbalanced at (60:T) 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.

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 <.

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