CSC130 Assignment 3

1. Inserting the keys in the order A, B, C, D, E, F into an initially empty BST gives a worst-case tree where every node has one null link, except one at the bottom, which has two null links. Give five other orderings of these keys that produce worst-case trees. **(5 Points)**

Answer: Here are 5 orderings and there are other correct orderings as well.

1) A, B, C, D, E, F,
2) A, F, B, C, D, E
3) A, F, B, E, C, D
4) A, F, B, E, D, C
5) F, E, D, C, B, A

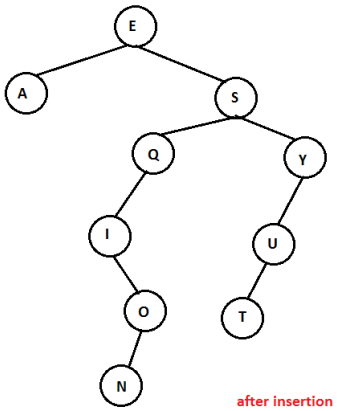
2. Give five orderings of the keys A, B, C, D, E, F that, when inserted into an initially empty BST, produce the best-case tree. **(5 Points)**

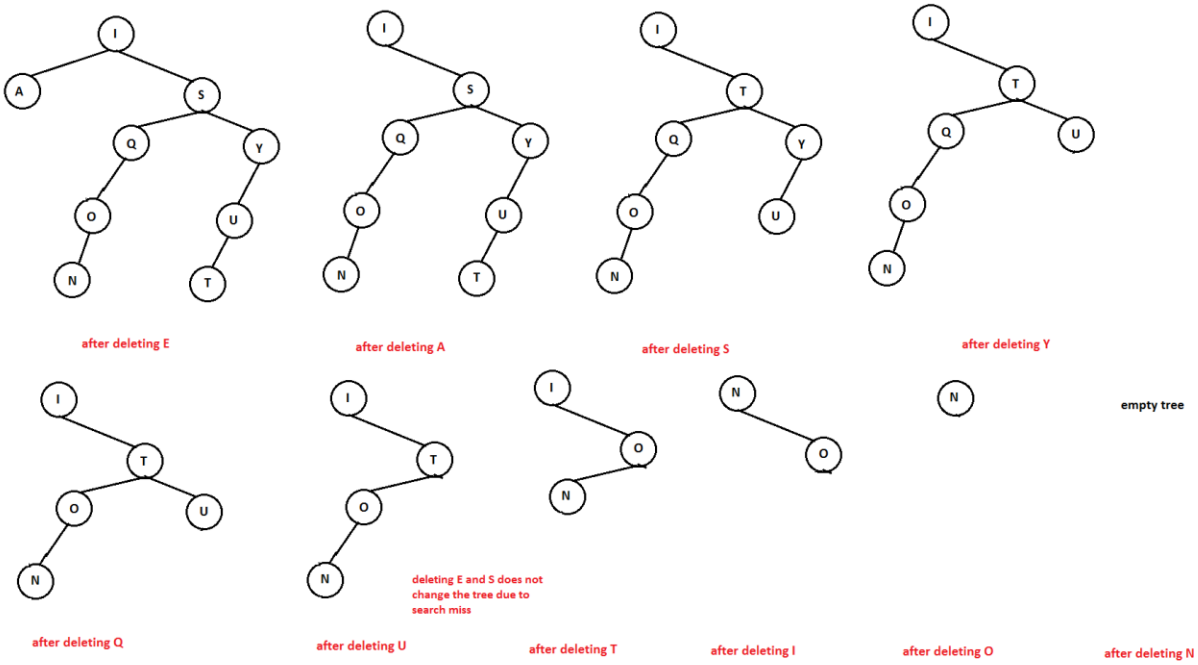
Answer: Here are 5 orderings and there are other correct orderings as well.

1) C, B, A, E, D, F
2) C, B, A, D, F, D
3) C, A, B, E, D, F
4) C, A, B, D, F, D
5) D, C, A, B, E, F

3. The BST is the result of inserting the keys, E A S Y Q U E S T I O N, in that order. Draw the sequence of BSTs that result when you delete the keys from the tree one by one, in the order they were inserted. **(10 Points. Ideally BST.delet(…) method should be followed here, but it is okay to use predecessor instead of successor to replace the removed node since the question does not make that clear.)**

Answer:





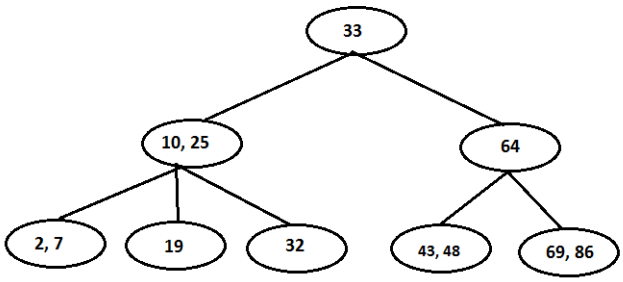
4. Suppose that a certain BST has keys that are integers in this array `[8, 22, 25, 28, 36, 39, 49, 50, 72, 90]`, and we search for 36. Which sequence below cannot be the sequence of keys examined? **(5 Points)**

a) 8, 90, 25, 72, 25, 50, 28, 49, 39, 36
b) 25, 28, 72, 36
c) 28, 90, 39, 25, 36
d) 90, 8, 39, 28, 22, 36
e) 90, 72, 50, 49, 39, 36

Answer: a), c), d)

5. Draw the 2-3 tree that results when you insert the keys `[64, 2, 33, 43, 25, 69, 10, 19, 48, 32, 7, 86]` in that order into an initially empty tree. **(5 Points)**

Answer:



6. Give the contents of a linear-probing hash table that results when you insert the keys E A S Y Q U T I O N in that order into an initially empty table of initial size M = 5 that is expanded with doubling whenever half full. Use the hash function c.hashCode() % M to transform Character c into a table index. **(10 Points. This one was not graded)**

Answer:

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
keys		Q		S	T	A	U			Y	E			I					N	O

7. Give non-recursive implementation of select() inside edu.csus.csc130.assignment3.BST.selectI(…). **(20 Points)**

8. Give non-recursive implementation of getHeight() inside BST.getHeightI(), which should take linear space and constant time per query. Hint. The implementation is similar to size(…). You should 1) add a field to Node class, 2) add a constructor to Node class taking the existing fields plus the new field as parameters, 3) make appropriate changes in methods where height can be changed. **(20 Points)**

9. Develop a LinearProbingHashSet implementation by starting with the code for LinearProbingHashST and eliminating all of the code involving values. **(20 Points)**

Submission Notes

1. For questions 1-6,
    a. Write your answers inside a text document (in plain text, Word, or PDF format) with the proper file extension
    b. Name the file as **firstname.lastname.assignment3.txt(docx or pdf)**
2. For questions 7-9
    a. Use JDK 1.8 and Junit4
    b. Put your full name at the beginning of every source file you created or modified. 5 points will be deducted if your names are not included in the source files.
    c. Do not change the provided package name or class name. You can add extra classes or methods if they are needed.
    d. If your code does not compile, you will get zero point.
    e. Use the provided tests to verify your implementation. Extra tests might be used for grading.
    f. Zip all the source files into **firstname.lastname.assignment3.zip**
3. Submit both of your files (text document and zip file) via My SacCT course web site.
4. Due Nov 28, 11:59 PM