CS206A Data Structure

HW5

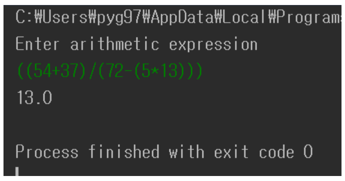
Student ID: 20160253

name: 박예기 (Park ye gi)

Problem1.

I made BinaryTree class. It has construct function which makes tree by using list. The form of the tree is [[level 0], [level 1], [level 2], …]. The BinaryTree class has function Evaluate which evaluates the answer of the arithmetic expression.

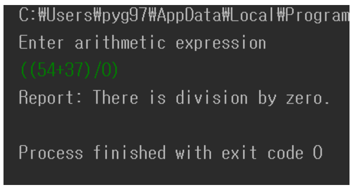
The result of evaluation is shown below.



```
C:\Users\pyg97\AppData\Local\Program
Enter arithmetic expression
((54+37)/(72-(5*13)))
13.0

Process finished with exit code 0
```

If there is division by zero in expression, it reports it and doesn't evaluate the answer.



```
C:\Users\pyg97\AppData\Local\Program
Enter arithmetic expression
((54+37)/0)
Report: There is division by zero.

Process finished with exit code 0
```

Problem2(a).

I think it is good to use pre-order traversal to add, remove, and find in binary search tree.

Pre-order traversal is firstly to process the root, secondly to process left subtree with recursive call, and thirdly to process right subtree with recursive call.

It can also be written in pseudocode below.

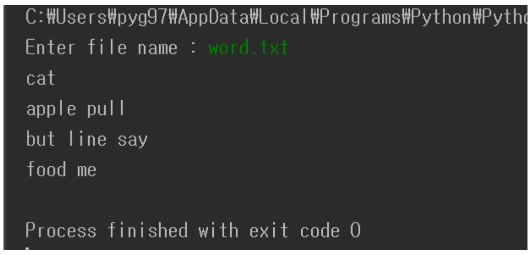
```
public void preorderPrint()
{ System.out.println(data);
   if (left != null)
      left.preorderPrint();
   if (right != null)
      right.preorderPrint();
}
```

Problem2(b).

In text file word.txt, the words are listed in the order of pre-order traversal.

Fisrt, I made BinarySearchTree class. It gets user input to select file to open (for example, word.txt). It firstly change the contents of file to list which is ordered in pre-order traversal and then changes that list to the form of the tree. The form of the tree is [[level 0], [level 1], [level 2], …]. I defined another function, def breadth_first_order(). This function print breadth first order of the tree.

The capture of result is shown below.

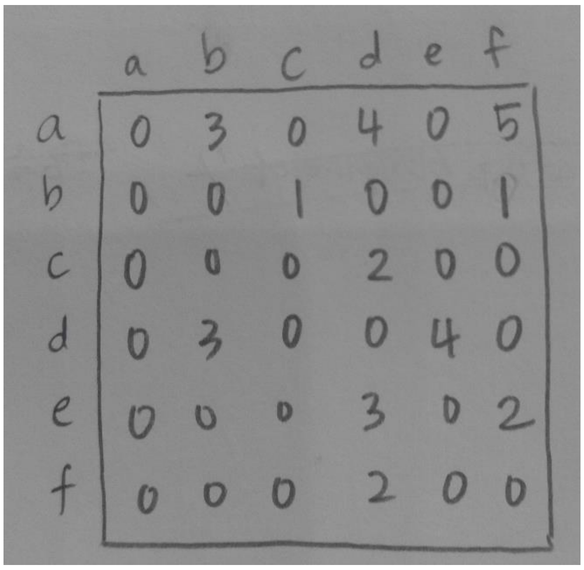


```
C:₩Users₩pyg97₩AppData₩Local₩Programs₩Python₩Pytho
Enter file name : word.txt
cat
apple pull
but line say
food me

Process finished with exit code 0
```

Problem3(a).

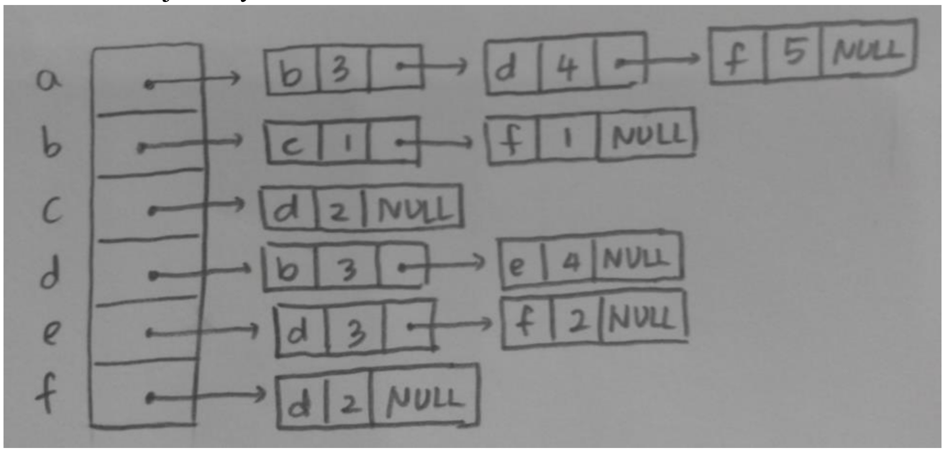
The adjacency matrix is shown below.



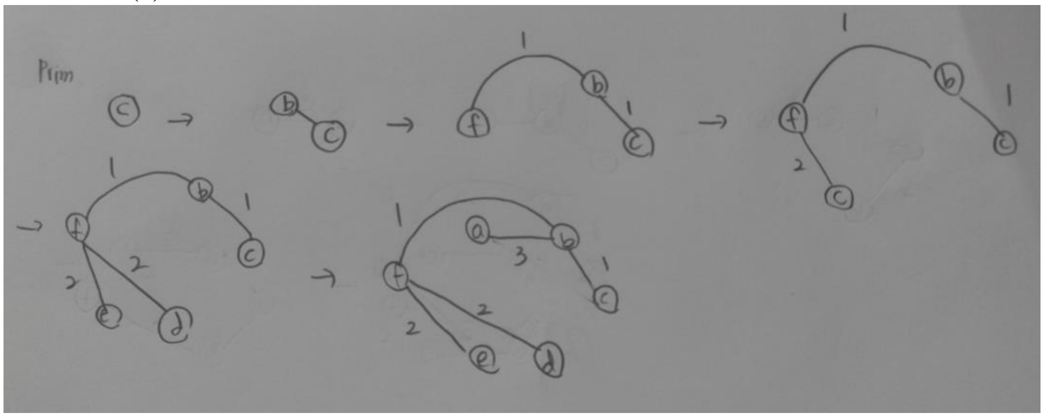
$$\begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & 0 & 3 & 0 & 4 & 0 & 5 \\ b & 0 & 0 & 1 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 2 & 0 & 0 \\ d & 0 & 3 & 0 & 0 & 4 & 0 \\ e & 0 & 0 & 0 & 3 & 0 & 2 \\ f & 0 & 0 & 0 & 2 & 0 & 0 \end{array}$$

Problem3(b).

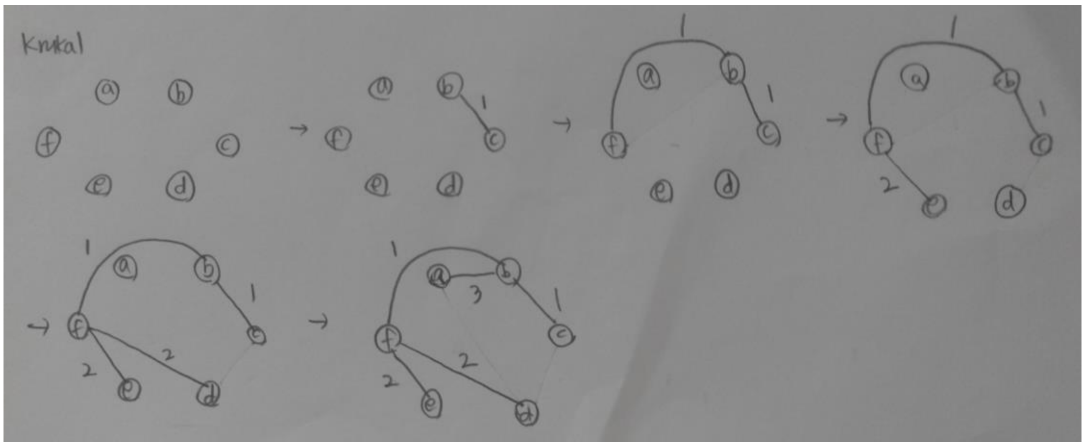
The linked adjacency list is shown below.



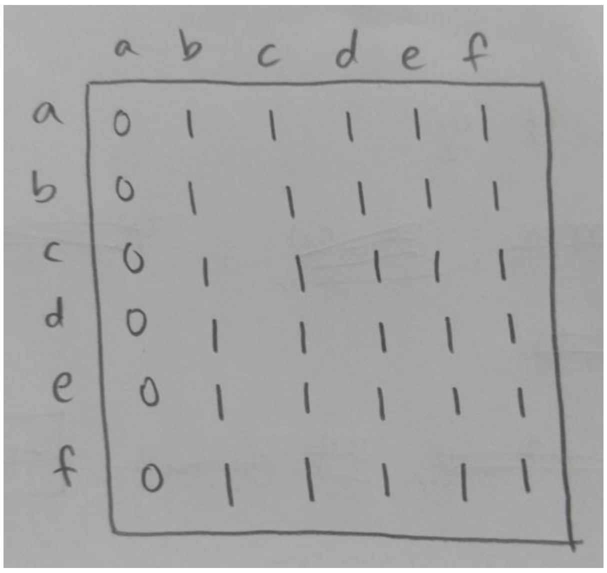
Problem4(a).



Problem4(b).



Problem5.



$$\begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & 0 & 1 & 1 & 1 & 1 & 1 \\ b & 0 & 1 & 1 & 1 & 1 & 1 \\ c & 0 & 1 & 1 & 1 & 1 & 1 \\ d & 0 & 1 & 1 & 1 & 1 & 1 \\ e & 0 & 1 & 1 & 1 & 1 & 1 \\ f & 0 & 1 & 1 & 1 & 1 & 1 \end{array}$$