Binary Search Tree Assignment

Write a java program (**using java.io.RandomAccessFile class)** that create, store and manipulate a set of ***n*** fixed-length records as binary search tree index on a **binary file**. Each record consists of four integers. The file at creation time (if n=8) should contain the following:

1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
-1	0	0	0

The first node (Node with index ZERO) will always have the index of the first empty node (at its first integer) in order to add a new record. Each empty node will have the index of the next empty node, forming a linked list of empty nodes.

The root will be always at node number 1

Your Program should handle the cases on insertion, deletion and In-Order traversal.

After insertion, each node should have 4 integers:

- First integer is the node value
- Second integer is the byte offset to record in a data file
- Third integer is the index of the left child of the node
- Fourth integer is the index of the right child of the node

Given the following insertions, you should insert the records IDs and the records references into the binary search tree index file.

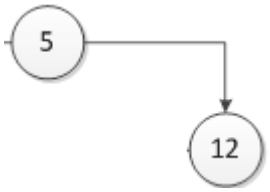
For Example, consider the following insertions:

Insert (Record ID: 5, Record Reference: 12)

The file should look as follow:

2	0	0	0
5	12	-1	-1
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
-1	0	0	0

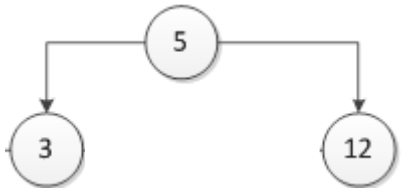
Insert (Record ID: 12, Record Reference: 24)



The file should look as follow:

3	0	0	0
5	12	-1	2
12	24	-1	-1
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
-1	0	0	0

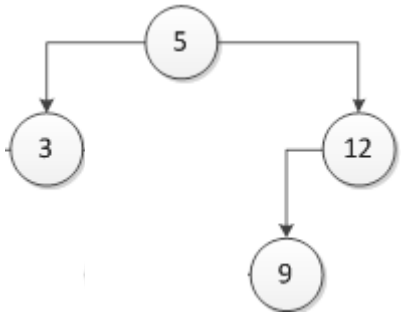
Insert (Record ID: 3, Record Reference: 36)



The file should look as follow:

4	0	0	0
5	12	3	2
12	24	-1	-1
3	36	-1	-1
5	0	0	0
6	0	0	0
7	0	0	0
-1	0	0	0

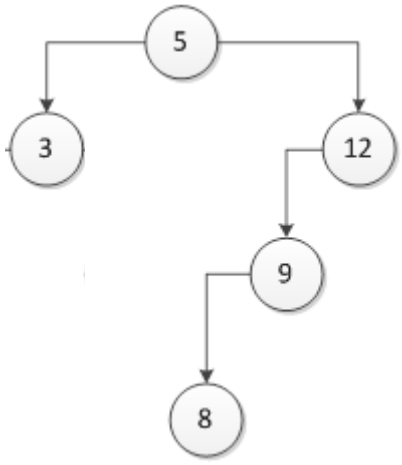
Insert (Record ID: 9, Record Reference: 48)



The file should look as follow:

5	0	0	0
5	12	3	2
12	24	4	-1
3	36	-1	-1
9	48	-1	-1
6	0	0	0
7	0	0	0
-1	0	0	0

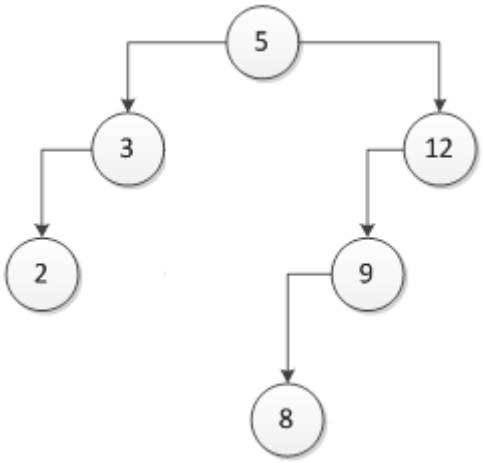
Insert (Record ID: 8, Record Reference: 60)



The file should look as follow:

6	0	0	0
5	12	3	2
12	24	4	-1
3	36	-1	-1
9	48	5	-1
8	60	-1	-1
7	0	0	0
-1	0	0	0

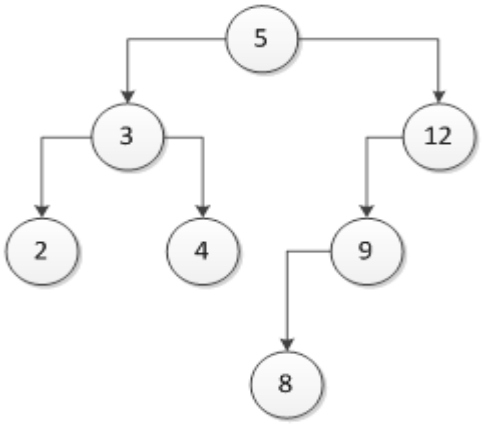
Insert (Record ID: 2 , Record Reference : 72)



The file should look as follow:

7	0	0	0
5	12	3	2
12	24	4	-1
3	36	6	-1
9	48	5	-1
8	60	-1	-1
2	72	-1	-1
-1	0	0	0

Insert (Record ID: 4 , Record Reference : 84)



The file should look as follow:

-1	0	0	0
5	12	3	2
12	24	4	-1
3	36	6	7
9	48	5	-1
8	60	-1	-1
2	72	-1	-1
4	84	-1	-1

The Requirements:

In order to deliver your program, you **must** use the following functions header:

void CreateRecordsFile (String filename, int numberOfRecords) (1 Grade)

int InsertNewRecordAtIndex (String filename, int Key, int ByteOffset) (3 Grades)

// Insert function should return -1 if there is no place to insert the record or the index where the new record is inserted if the record was inserted successfully.

void SearchRecordInIndex (String filename, int Key) (1.5 Grades)

// This method should return the byte offset of the record or -1 if the record is not found

void TraverseBinarySearchTreeInOrder (String FileName) (1 Grade)

// This method will traverse the binary search tree with the "In Order" method

void DisplayIndexFileContent (String filename) (0.5 Grade)

// This method should display content of the file, each node in a line.

Notes:

- **Assignment Delivery** On the week starting **14/4** on your lab
- Assignment should be done **On a Group of 2 students or individually only**.
- All the submitted code must be completely yours. Your grade depends on delivering a running code and your understanding for the code.
- You need to deliver a complete app with the ability to execute the mentioned methods during the runtime.
- For any questions kindly contact "a.abdelaziz@fci-cu.edu.eg"