CSE 2331
(Binary Tree Programming Assignment)
Autumn, 2017

THIS PROGRAMMING ASSIGNMENT SHOULD BE DONE INDIVIDUALLY. You may discuss the algorithm with other students BUT DO NOT LOOK AT OR COPY ANYONE ELSE'S CODE.

Write a program to read a set of integers from a file, dataX, and a set of ranges from a second file, rangeX, and, for each range $[a, b]$ in rangeX, report the if the sum of all the integers in dataX which are in the range $[a, b]$ is even. As the integers are read from file dataX, insert them in a binary search tree. After all the integers have been inserted into the binary search tree, read the ranges from file rangeX and query the binary search tree to report the if the sum is even. The program needs to do this *without* calculating the actual sum of numbers in the range (for example, given that both a and b are odd, you can determine that $a + b$ is even without knowing their respective values).

For instance, if file dataX has the integers $(3, -2, 2, 5, 2, -6, 1)$ and file rangeX has the ranges $([-3, 4], [0, 5])$ then the output should be:

```
Range [-3,4]. even sum
Range [0,5]. odd sum
```

The insert and queries should run in $O(h)$ time (for each insert or query) where h is the height of the tree. You need to insert elements in the tree in the order they are read from the file but you do not need to do any balancing/rotations on the tree. (Because the integers are "randomly" ordered in the file, the expected height of the tree is $O(\log_2(n))$.)

1. Describe how to modify the binary search tree data structure to support computing if the sum in a range is even in $\Theta(h)$ time.
2. Describe the algorithm to insert elements in your modified binary search tree and GIVE PSEUDO-CODE for your insertion algorithm.
3. Describe the algorithm to report if the sum of the elements in range $[a, b]$ is even and GIVE PSEUDO-CODE for your reporting algorithm.
4. Implement your algorithm in Java, C++ or C, implementing the binary tree from scratch. You may not use the binary tree components from the OSU java components used in Software I or II). (Program specifications are given below.)
5. Test your program on sample data files in Carmen and make sure it runs on stdlinux.

Program specification:

- Your program may be written in Java, C++ or C. You may not use the binary tree components from the OSU java components used in Software I or II, you should implement the binary tree from scratch.
- To use java in stdlinux, type "subscribe" and then select JDK-CURRENT. To compile evenSumRange.java in stdlinux, type "javac evenSumRange.java". (Use the javac command to compile other files/classes.) To run the program evenSumRange, type "java evenSumRange dataX rangeX".
- Name your program `evenSumRange`.
- Your program has two command line arguments, dataX and rangeX, representing an input data file name and an input range file name. A user runs your program on files dataX, rangeX, by typing:

  ```
  evenSumRange dataX rangeX
  ```

  Sample dataX and rangeX files are in the Carmen.
- Output from your program should be a list of ranges and wheter the sum of the integers in each range is even. Each range and sum should be on a separate line in the output.

- File dataX consists of of integers, one per line. Read the file until reaching the end of file.
- File rangeX consists of pairs of integers, one pair per line. Read the file until reaching the end of file.
- Your program MUST insert the integers from dataX in a binary search tree in the order they are read from dataX.
- Your program should have a function/method Function `Insert` (or `btreeInsert`) for inserting an integer in the binary search tree. `Insert` (x) inserts integer x into the binary search tree. `Insert` (x) should run in $\Theta(h)$ time where h is the height of the tree.
- Your program should have a function/method `evenSumRange` (or `btreeEvenSumRange`) for computing if the sum of all the elements in the btree which lie in the range is even. Function `evenSumRange` (a, b) returns **true** the sum of all the elements x in the binary search tree where $a \leq x \leq b$ is even. `evenSumRange` (a,b) should run in $\Theta(h)$ time where h is the height of the tree.
- Your program should NOT do any additional processing between the insertions and the `evenSumRange` queries. In particular, your program should NOT be doing any $\Theta(n)$ processing after the insertions.
- In Java, store the integer values in an integer of type "`long`". In C or C++, store the integer values in an integer of type "`long long`".
- Test your programs on the data files in Carmen.
- Your program may be written in C, C++ or java. All coding for this lab is to be done individually. You may discuss this lab with other students but DO NOT LOOK AT OR COPY anyone else's code.
- Check that your program runs and compiles on stdlinux. The grader will compile and test the program on stdlinux. Programs which do not compile on stdlinux will receive 0 points.

In the dropbox on carmen submit:

1. A plain text or pdf file named README giving instructions on how to compile and run your program. (These may be trivial if your program is a C++ program contained in a single file. If your program needs to link libraries, be sure to include these in the instructions).
2. A plain tex or pdf file containing:
   (a) A description of the modifications to the binary search tree to support computing `evenSumRange` in $\Theta(h)$ time;
   (b) A description of the algorithm to insert elements in the modified binary search tree and PSEUDO-CODE for the insertion algorithm;
   (c) A description of the algorithm to report the sum of the elements in range $[a, b]$ and PSEUDO-CODE for the reporting algorithm.
3. Source code (NOT object code) for your program.

Do NOT submit object code and certainly DO NOT submit any data files. The grader will compile and run your program.