Assignment 7: Data Structures

CS 69001: Computing Lab 1

Introduction:

In this assignment you have to implement various operations related to Interval Tree and Interval Graphs.

Objective:

The aim of this assignment is to get acquainted with the Interval Tree and Interval Graph data structure.

Problem Description:

Interval Tree:

A tree data structure to hold intervals is called interval tree. It is used to find all the intervals that overlaps with any given interval. Trivially the operation of finding intervals requires O(*n*) time, where *n* is the total number of intervals. Time complexity of this naïve approach is asymptotically optimal because a query may returns all the intervals in the collection. For example, in a case where the query is a large interval intersection all the intervals in the collection.

However, output-sensitive algorithms can be considered to get a better time complexity. In this context, output-sensitive algorithms will be the ones whose runtime is dependent on the number of overlapping intervals for a particular query. Centered interval trees can be used to obtain a query time of O(log n + m) where, m is the number of overlapping intervals.

Please go through the following link to get an understanding of what a centered interval tree is.

http://www.dgp.toronto.edu/people/JamesStewart/378notes/22intervals/

Interval Graph:

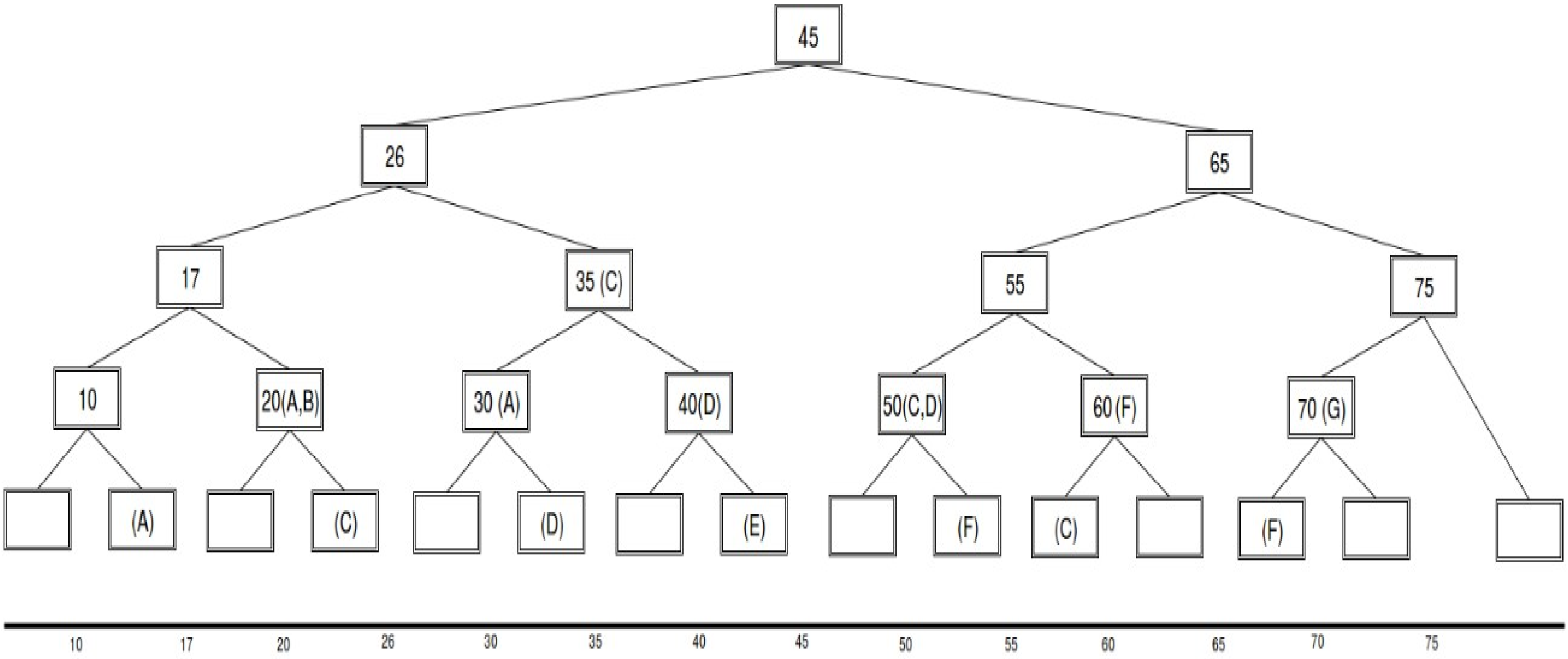
An interval graph is an undirected graph where each node can be put into one-to-one correspondence with a set of intervals on the real line. There is an edge between every pair of vertices whose corresponding intervals have non empty intersection.

Example:

For instance, let us consider the following set of intervals.

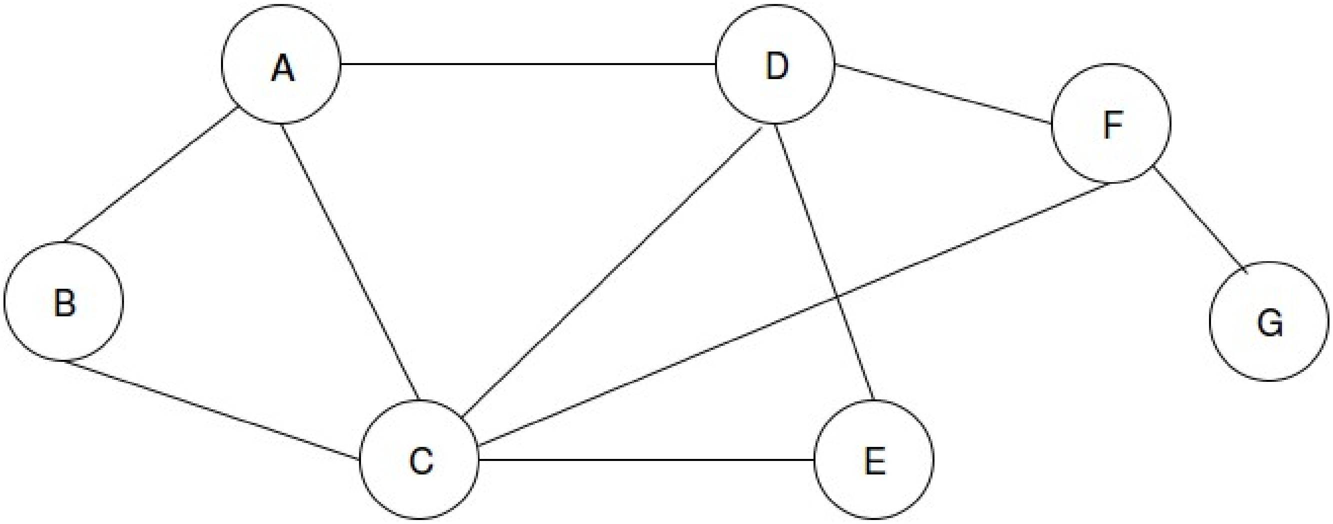
A	10 - 35
B	17 - 26
C	20 - 60
D	30 - 55
E	40 - 45
F	50 - 70
G	65 - 75

The set of intervals can be represented using a centered interval tree as follows.



Here integer stored in each internal nodes represents the end points of the intervals and the alphabets stored in the nodes represents the intervals.

Following is the corresponding interval graph.



Problem Statement:

Suppose a music theorist is studying similarities and differences between the style of composing of different composers across various periods. He is given with a database of large number of musicians with the songs composed of different genres in a CSV file, a simplified database is of the format as given in the table below.

Sl No.	Composer Name	Lifespan	genre1	genre2	genre3	genre4	. . .
A	C1	1788-1871	Song1, Song2, Song3, Song4	N.A.	Song5, Song6, Song7	Song8,Song9, Song10	. . .
B	C2	1774-1851	Song11, Song12, Song13	Song14,Song15	N.A	N.A	. . .
C	C3	1743-1807	N.A.	Song16,Song17, Song18	Song19,Song20Song21,Song22	N.A	. . .
D	C4	1679-1728	Song23, Song24	Song25,Song26	Song27,Song28, Song29	Song30	. . .
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .

You need to built a tool using Python for the music theorist to perform various operations on the database. To easily conduct the operation, first the Sl No. and the Lifespan of the composer is represented using a Interval tree. After a particular query is performed an the output is obtained, rest of the informations are retrieved from the CSV file with reference to the output. Following are the operations you need to implement.

Basic operations (20 + 20 marks)

- BuiltIT(): Given a CSV file built a Centered Interval Tree. *SL. No.* and *Lifespan* should be stored in each node. Generate the input file for *Graphviz* to display the constructed tree.

- Insert(): Given *Sl No.* and *Lifespan* of a composer, insert it into the Interval Tree. Generate the input file for *Graphviz* to display the constructed tree. Assume that endpoints of the given lifespan is present in the constructed tree.

Query of intervals (10 + 10 + 10 marks)

- FindOverlaps(): Given a close year interval, find the composers whose lifespan overlaps that of the input, print their *Sl No.* and *Lifespan.*

- MaxOverlaps(): Given a close year interval, print the *Sl No.* and *Lifespan* of the composers having maximal overlap in their lifespan.

- FindSongs(): Given a close year interval, print the songs composed of different genre in that interval. Use a look up table to retrieve songs efficiently.

Interval Graph (10 + 10 + 10 marks)

- BuiltIG(): Use Interval Tree to construct a Interval Graph/ Forest. Generate the input file for *Graphviz* to display the constructed tree.

- BiggestTeam(): Find the biggest team of composers who might have worked together in a group for certain period of time. Also, display the time period in which they might have worked together.

- FindGroup(): Find the biggest and the smallest group of composers in which each must have overlaps in their lifespan with atleast one in the group. Generate the input file for *Graphviz* to display the constructed tree.

Important Note
You can not import any package in your code. Marks will be not be given if you do so.

Deliverables:

You will submit your Python programs and a note explaining your implementation and the time complexity of the algorithms used in a single rollno_a7.tar.gz file in the moodle submission link. It is mandatory that your code should follow proper indentation and commenting style. There will be deductions in the awarded marks, if you fail to do so.