

CS20B: Homework/Project 5 (160 points)

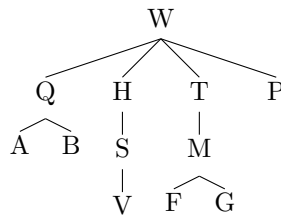
prof. dr. Irma Ravkic

This homework covers some questions about binary search trees and graphs. Please do the questions individually (without your coding partners), and only the coding exercises with your partner. The format of your submission will be the same as before.

1 Questions (70 points)

In your answer document please include the question text as well so I can easily match questions to your answers.

1. (15 points) Answer the questions below about the following tree, using the letters that label the nodes:



- (a) (1 point) Which node is the root?
 - (b) (1 point) Which node is the parent of F?
 - (c) (1 point) Which node(s) are the children of H?
 - (d) (2 points) Which node(s) are the ancestors of M?
 - (e) (2 points) Which node(s) are the descendants of T?
 - (f) (3 points) Which node(s) are the siblings of S? And of H?
 - (g) (1 point) What is the height of the tree?
 - (h) (2 points) Write the order in which the nodes are visited by a breadth-first traversal. Use the following format when specifying the order: [node1 node2 ... nodeN]
 - (i) (2 points) Write the order in which the nodes are visited by a depth-first traversal. Use the following format when specifying the order: [node1 node2 ... nodeN]
2. (20 points) Answer the following questions about the binary search tree (in Figure 1).
- (a) (2 points) What is the height of the tree?

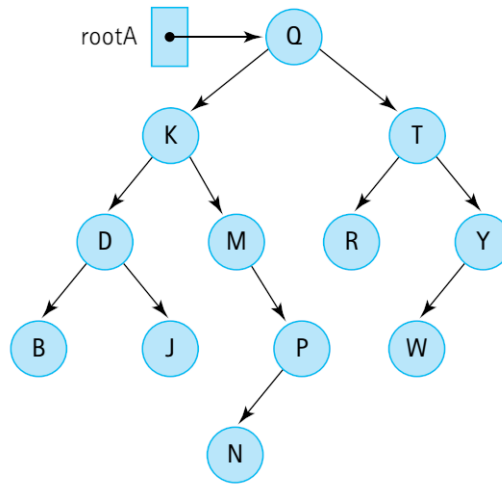


Figure 1: A binary search tree

- (b) (2 points) Which nodes are on level 3?
 - (c) (2 points) Which levels have the maximum number of nodes that they could contain?
 - (d) (2 points) What is the maximum height of a binary search tree containing these nodes (counting level 0 as well)?
 - (e) (2 points) What is the minimum height of a binary search tree containing these nodes (counting level 0 as well)?
 - (f) (2 points) What is the order in which nodes are visited by a preorder traversal?
 - (g) (2 points) What is the order in which nodes are visited by an inorder traversal?
 - (h) (2 points) What is the order in which nodes are visited by a postorder traversal?
 - (i) (2 points) What is the order in which nodes are visited by a depth-first traversal?
 - (j) (2 points) What is the order in which nodes are visited by a breadth-first traversal?
3. (10 points) **(Hashing from Chapter 8)** Critique the following hash functions for a domain consisting of people with attributes *firstName*, *lastName*, *number* (used to resolve identical first and last names, e.g., “John Smith 0,” “John Smith 1,” etc.), and *age*. The names are of class String and the other two attributes are of type int.
- (a) hash function returns $(age)^2$
 - (b) hash function returns $(age)^2 + \text{lastName.hashCode}()$
 - (c) hash function returns $\text{lastName.hashCode}() + \text{firstName.hashCode}()$

- (d) hash function returns `lastName.hashCode() + firstName.hashCode() + number`

Note: answer the question in terms of how good the hashCode is for avoiding collisions (different entries having the same hashCode) and uniquely identifying the objects.

4. (5 points) (**Hashing from Chapter 8**) Critique the following code which is intended to print out whether or not a key value *k1* is used in a map named *relationships*:

```
1  if (relationships.get(k1) != null)
2
3      System.out.println("yes it does");
4
5  else
6
7      System.out.println("no it does not");
```

Now rewrite the code so that it works correctly

5. (10 points) (**Graphs**) Use the following description of an **undirected graph** for Questions a) and b) below:

EmployeeGraph = (V, E)

V(EmployeeGraph) = {Susan, Darlene, Mike, Fred, John, Sander, Lance, Jean, Brent, Fran}

E(EmployeeGraph) = {(Susan, Darlene), (Fred, Brent), (Sander, Susan), (Lance, Fran),

(Sander, Fran), (Fran, John), (Lance, Jean), (Jean, Susan), (Mike, Darlene), (Brent, Lance), (Susan, John)}

- (a) Draw a picture of EmployeeGraph (draw it the way graphs look like - see the book or slides)
- (b) Which one of the following phrases best describes the relationship represented by the edges between the vertices in EmployeeGraph and **why**?
- “works for”
 - “is the supervisor of ”
 - “is senior to”
 - “works with”
6. (10 points) (**Graphs**) Draw the adjacency matrix for the graph in Figure 2. Store the vertices in alphabetical order.

2 Programming exercises (90 points)

Please do these with your partner. The assignments are not precisely defined (as in the real-world) and you will have to figure out the details. You will need to

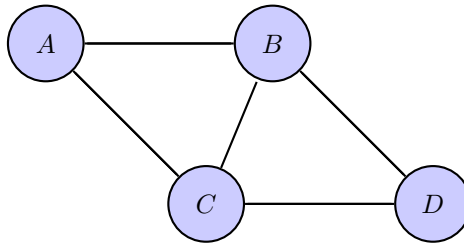


Figure 2: Graph example

show the concepts you learned in this class (OOP design) as much as possible. You will present your solution to the class in the final class of the semester. You will run your code and demonstrate how it works and then we will discuss the structure of your solution. I will ask each of you to write a report on your partner describing how much your partner contributed to this homework. Make sure you comment your code.

2.1 Problem 1: Logical Circuits (45 points)

In this assignment using Java's OOP design you need to represent the following logic gates: AND, OR, NOT and XOR. The basic functionality of your code should be the following:

1. For each gate, given some inputs, get the calculated output. For example (this is pseudocode): `AND(True, True)` should give me `True`, `XOR(True, False)` should give me `True`. Make the application that makes sure **each** of your gate operates correctly with all combinations of binary inputs.
2. (Bonus for A+) **Go for the glory!** This one is a bit more complicated but fun. Given a logic circuit calculate the output of the circuit. For example, for the circuit in Figure 3 and inputs `A = TRUE`, `B = FALSE`, `C = FALSE`, `D = TRUE` the circuit outputs `FALSE`. Don't try to solve everything. Just make a solution for these simpler examples. Use the circuit in Figure 3 to test your code.

You can always do more. Amaze us.

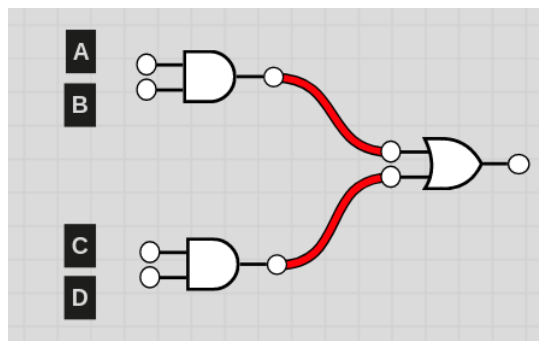


Figure 3: Example circuit

2.2 Problem 1: Social Network (45 points)

You are provided with the following two files: *users.csv* and *edges.csv*. The files contain the following:

1. *users.csv* has the following attributes:
 - **user_id**: is a unique identifier of a user
 - **community**: is one of the following values {science, sport, art}.
 - **age**: is the age of a user
 - **posts**: is a list of post types users most frequently share (to simply it for you I set maximum possible per user to 10). It can be one of the following values: {video, image, article}.
2. *edges.csv* has the following attributes:
 - **user_id, user_id** : represents the edge between two user ids

2.2.1 What should I do?

To solve this task use and adapt the WeightedGraph implementation from the book (ch10). Represent this data as a graph and perform some interesting analysis of the data. You should demonstrate at least three questions asked about the graph. Some of these will require adding methods and new attributes to WeightedGraph.java file or performing it in the driver code.

2.2.2 What is a CSV file?

CSV stands for “comma-separated values”. You can load these files in Excel or Libre Office Calc and view them. However, for this project you will need to load it in Java.

2.2.3 Visualisation

If you would like to visualize your graph (or a part of it) I suggest using the following library for representing graphs: <http://graphstream-project.org/download/> Go to the Website and download all the three jars from the 1.3 release (gs-core, gs-algo, gs-ui). Make sure to download them somewhere you can find them. I suggest putting all three jars in a folder on the same level as your *src* folder, and call it *lib*. In your IDE make sure you add all three jars to your library list (Configure Build Path -> Add external jars). You have to figure out how to use the library. :)

2.2.4 Timely report errors or issues

If you find an inconsistency in the data, code or similar, report immediately. It's also part of the process and could bring you even more glory. :)