# **20 Questions: Binary Tree Data Structure**

Suppose you want to write a program to play a game of 20 questions. In this lab you will design the data structures and set them up to play such a game. Consider the following image:

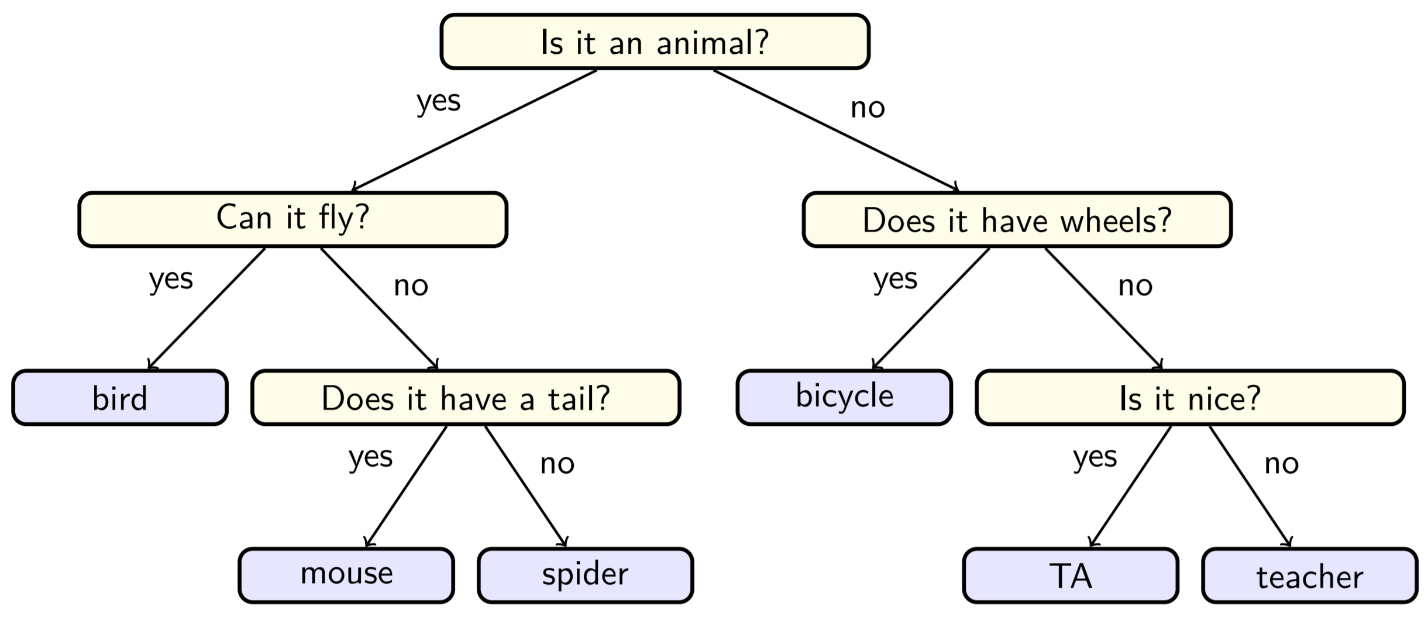


Image source: <https://courses.cs.washington.edu/courses/cse143/20sp/assessments/20-questions/>

(Be careful about using the washington.edu webpage. I am structuring this assignment differently.)

Each node in the above tree will be a single Java object of type QuestionNode. For example, “Is it an animal?” is going to be an object. The QuestionNode class will have at least the attributes and methods described in the following UML diagram (but you’re allowed to add more if they are appropriate).

|  |
| --- |
| QuestionNode |
| - question : String  - yes\_answer : QuestionNode  - no\_answer : QuestionNode |
| + <<constructor>> QuestionNode(question : String)  + isTerminatingQuestion() : boolean  + ask() : void //print the question  + setNextQuestion(q : String, boolean is\_yes) : void  + getNextQuestion(boolean is\_yes) : Question |

**question** is a String variable containing the text of the question, such as “Is it an animal?”. When the node is actually a final answer, the **question** variable contains the answer, such as “bicycle”.

**yes\_answer** is a QuestionNode variable containing a reference to the next question to ask when the user’s answer to the question is “yes”. The **yes\_answer** variable is ***null*** if the node is a final answer.

**no\_answer** is a QuestionNode variable containing a reference to the next question to ask when the user’s answer to the question is “no”. The **no\_answer** variable is ***null*** if the node is a final answer.

The **constructor**  should initialize the **question** attribute to the given input value and initialize the **yes\_answer** and **no\_answer** variables to ***null***.

The **isTerminatingQuestion** method returns true if this node is a final answer, which is indicated by the **yes\_answer** and **no\_answer** variables referring to ***null***. This method returns false otherwise.

The **ask** method prints out the node’s question.

The **setNextQuestion** method takes a String **q** and a boolean **is\_yes** as input. The method creates a new QuestionNode with the question **q** and sets the QuestionNode as the value of the **yes\_answer** variable if **is\_yes** is true. Otherwise the **no\_answer** variable is set to this new QuestionNode.

The **getNextQuestion** returns the value of **yes\_answer** if **is\_yes** is true. Otherwise **no\_answer** is returned.

In addition to writing the above Class, you must write a main method that creates and composes a 20 Questions tree at least as complicated as the one in the above image.

In other words, your tree must have at least 5 question nodes with all the answers filled in.

Your main method must also allow the user to play the game to completion.

You can earn up to 98% for completing all of the above. This is what you should shoot for first.

In order to earn that last 2%, you must write your code so that when the user wins and the binary tree is stumped, you prompt the user to enter an extending question with two valid answers and you extend the tree so that the user can play again with a larger and “smarter” tree.

For example, consider the following example interaction. User input is in red. Program output is in blue:

Is it an animal?yes

Can it fly?yes

bird?no

At this point the program might ask the following:

Please give me a question to ask at this point:Does it have feathers?

If the user says yes, what should the answer be?bird

If the user says no, what should the answer be?bat

Then the given information will be used to replace the answer “bird” with a new QuestionNode in the tree.