CSCI 204 – Introduction to Computer Science II

Lab 9 – 20 Questions a with Binary Tree

1 Objectives

The objectives of this lab are to:

- Become familiar with binary trees
- Know how to traverse a binary tree
- Learn how to modify a binary tree
- Use recursion to traverse and modify a binary tree

Read the entire lab description before you begin working on the assignment.

2 Introduction

In this lab you will write a program for playing an interactive computer game called *Animal* (a variant of the game "20 Questions"). The computer plays by trying to guess the name of an animal the player imagines. Although an animal may have many characteristics, considering only one at a time— "Is the animal furry?" or "Does it have horns?" —reduces its description to a series of binary (two way) choices. Some sample output from a run will help you understand the game. The human types only "yes" or "no" for each question that the program poses. In the following transcript, the bold-faced letters are typed by the human user. The rest are generated by the program.

```
Think of an animal.
Does it have fur? yes
Does it have horns? no
Is it a lion? No
```

You can see a full sample run of the game at the end of this document.

3 Storing data in a binary tree

The program uses a binary tree to hold the questions and the answers (animal names.) The interior nodes of the tree will always store the questions, and the leaves will always contain animal names.

The program begins at the binary tree's root and asks the question stored as a string in that node. Whether the left or right node is visited next depends on the answer. If the answer is a "yes" we take the right link, otherwise we take the left. We either reach a further question (an interior node) or a leaf (final node with an animal in it).

The program learns as it plays. If it reaches a leaf and guesses wrong, it asks the user questions about where it went wrong and expands the tree to hold the new knowledge.

For example, the following interaction might take place:

```
Is it a cow? no
I guessed wrong.
What animal were you thinking of? sheep
What question should I have asked to tell your sheep and my cow apart?
Does it bah?
Is the correct answer yes or no? yes
```

Internally, two new nodes are added to the tree after the above conversation, along with the implication that cows don't say *bah*. The new animal and the current node's animal are placed in the two new leaves. The new question is placed in the current node and the node changes from one that contains an animal to one that contains a question.

4 Printing the tree

Running the program in debugging mode asks the program to print the existing game tree after each round completes. The tree is printed from left to right with the root on the left, instead of a more common representation that places the root at the top. Each question is an interior node and the animals are the leaves. Here is an example of the output from debugging mode.

```
      +--chicken
      |
+--Does it fly?
|     |
|     +--emu
|
|
Does it have feathers?
|
|
|     +--giraffe
|     |
+--Does it have a long neck?
      |
      +--tiger
```

This can help you debug the program. When you complete your program correctly, you can run the program without the debugging mode. Currently the printing method is called in the `main()` method within the loop. You may add it somewhere else if you need to.

5 Getting started

Begin by making a working directory for this lab and copy the starter file `animalgame.py` from the following directory.

~csci204/2017-fall/student/labs/lab09/

This file gives you a starting point for your lab work. `animalgame.py` is a skeleton file where some basic functionality is already provided for you, such as printing the tree. You will need to complete the rest of the program so that the program behaves similar to the specifications.

6 How to proceed

The basic idea is that you need to traverse the existing tree starting from the root. At any moment in the program, the program examines the content of the current node, and traverses to either the left branch or the right branch until it reaches a leaf, based on the answer(s) from the user. If the program reaches at a leaf node (How do you determine if a node is a leaf node?), then the node contains the name of an animal. The program should print this name and ask the user if this is the correct answer. If it is not, the algorithm needs to expand this node into a new tree branch (three nodes, one with the question, the other two are child leaf nodes). If it is the correct answer, this round of the game is finished. If the current node is an interior node, it means the node contains a question. Your algorithm needs to print the question and ask

an answer from the user. If the answer is a 'yes', the program follows the right branch, otherwise follow the left branch.

7 Modify the `play()` method

You should complete the `play()` method. Overall, your strategy must be recursive. Note that the `play()` method is first called from a `while` loop within the `main()` method with the parameter as the 'root' of the tree. The logic of the `play()` method has been described in the above section (*How to proceed*.) You may write more methods as needed. This method will be similar to the ones for insert and search for binary search trees. Again, your solution **has to be recursive.**

Sketch of the algorithm for `play(node)`

1) *If it is a leaf*
    a) *Make a guess*
    b) *If you guessed it right*
        i) *Then display congratulatory message to the user and exit play method*
        ii) *Else add information to the tree*
2) *Else (it is an internal node)*
    a) *Ask the question*
    b) *If the answer is yes*
        i) *Then, follow the right branch*
        ii) *Else, follow the left branch*

8 Test your program

You must test your program at every step of the development, making sure it is working correctly. When all completed, run the program either in Idle, or at the command line.

9 Cool Extras

These features are extra credit.

1. Python is able to save your tree structure as a binary file and read it in when the game starts using a feature called *pickling*. Read about pickle on python's website. Modify your game so that it reads in a saved tree at the start (if one exists) and writes the memory at the end.

2. To help you out, we provide a function to print the tree periodically. Unfortunately, it was easier to print the tree sideways. Write a method to print the tree right side up. Call your method instead of ours. (Your method is traditionally called pretty-print).

10 Submission

Submit your `animalgame.py` to Moodle before the usual lab deadline. Double check to make sure your file(s) all got uploaded successfully.

11 Transcript of a sample run

Here is a full sample run of the game. It demonstrates growth of the tree in all directions.

```
Print the tree as you play? yes
Think of an animal.
Does it have feathers? yes
Is it a chicken? yes
Wow! I guessed it!
     +--chicken
     |
Does it have feathers?
     |
     +--tiger
Play again? yes
Think of an animal.
Does it have feathers? no
Is it a tiger? yes
Wow! I guessed it!
     +--chicken
     |
Does it have feathers?
     |
     +--tiger
Play again? yes
Think of an animal.
Does it have feathers? yes
Is it a chicken? no
I guessed wrong.
What animal were you thinking of? emu
What question should I have asked to tell your emu and my chicken
apart? Does it fly?
Is the correct answer yes or no? no
          +--chicken
          |
     +--Does it fly?
     |    |
     |    +--emu
     |
     |
Does it have feathers?
     |
     +--tiger
Play again? yes
Think of an animal.
Does it have feathers? yes
Does it fly? yes
Is it a chicken? no
I guessed wrong.
What animal were you thinking of? duck
```

```
What question should I have asked to tell your duck and my chicken
apart? Does it quack?
Is the correct answer yes or no? yes
                +--duck
                |
           +--Does it quack?
           |    |
           |    +--chicken
           |
           |
      +--Does it fly?
      |    |
      |    +--emu
      |
      |
Does it have feathers?
      |
      +--tiger
Play again? yes
Think of an animal.
Does it have feathers? no
Is it a tiger? no
I guessed wrong.
What animal were you thinking of? snail
What question should I have asked to tell your snail and my tiger
apart? Does it have a shell?
Is the correct answer yes or no? yes
                +--duck
                |
           +--Does it quack?
           |    |
           |    +--chicken
           |
           |
      +--Does it fly?
      |    |
      |    +--emu
      |
      |
Does it have feathers?
      |
      |
      |    +--snail
      |    |
      +--Does it have a shell?
           |
           +--tiger
Play again? yes
Think of an animal.
Does it have feathers? no
Does it have a shell? no
Is it a tiger? no
```

```
I guessed wrong.
What animal were you thinking of? anteater
What question should I have asked to tell your anteater and my tiger
apart? Does it eat ants?
Is the correct answer yes or no? yes
                +--duck
                |
           +--Does it quack?
           |    |
           |    +--chicken
           |
           |
      +--Does it fly?
      |    |
      |    +--emu
      |
      |
Does it have feathers?
      |
      |
      |    +--snail
      |    |
      +--Does it have a shell?
           |
           |
           |    +--anteater
           |    |
           +--Does it eat ants?
                |
                +--tiger
Play again? no
```