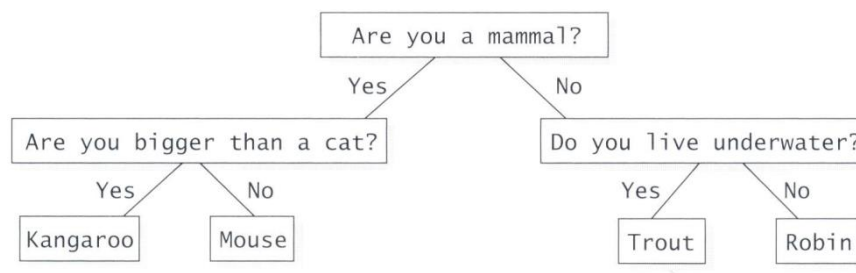


Project 2 - User Friendly Animal Guessing Program

Sungwon Kang

Due May 7

In this project, we are going to implement variations of the animal guessing program. When the program starts execution, it reads in the data from the data file and constructs a knowledge tree such as the following:



The program uses the knowledge tree to ask questions in order to guess the animal the user has in mind. For example, the following is an example of a dialogue between the program and the user:

Program: Are you a mammal? [Y or N]
User: Y
Program: Are you bigger than a cat? [Y or N]
User: Y
Program: My guess is Kangaroo. Am I right? [Y or N]
User: Y
Program: Shall we play again? [Y or N]
...

There are two formats to store data in the data file: A format and B format. The data file using the A format is named "data-A.txt" and the data file using the B format is named "data-B.txt". A format and B format are as illustrated below.

For the tree above, in the case of A format, the data are stored as follows (postorder traversal):

```
Kangaroo
Mouse
Are you bigger than a cat?
Trout
Robin
Do you live underwater?
Are you a mammal?
```

and in the case of B format, the data are stored as follows (preorder traversal):

```
Are you a mammal?
Are you bigger than a cat?
Kangaroo
Mouse
Do you live underwater?
Trout
Robin
```

Your program works interactively with the user and depending on whether the user prefers A format or B format, it should read in the (equivalent) data from "data-A.txt" or "data-B.txt" file, respectively. When your program terminates, your program should store the updated knowledge tree in both the "data-A.txt" file and the "data-B.txt" file.

Your program should implement two data structures that will work efficiently with the two data formats explained above and use one of the two the data structure depending on the user choice of data format. You can assume that once the user designates data format at the beginning of an interaction, the user cannot change it until the program finishes.

Part I. Implement a variation of the Animal Guessing game as explained above.

Part II. Compare the complexity of the two data structures with respect to each operation and performance of reading data from and storing data to the data files.

Part III. For this part, you need to make an enhancement over the version you implemented in Parts I and II, which is that your program tolerates the user's mistakes. That is, it is possible for the user to make mistakes while answering yes or no questions and so later when the user discovers that she has made a mistake she can go back (step by step) one step at a time until she met the question where a mistake was made. Then she can continue from that point on as usual.

So if the user enters "U" (for undo) instead of "Y" or "N", then the program should recognize that the user has made a mistake and ask a question like the following:

Is the question "Are you bigger than a cat?" where you made a mistake?

The following is an example of sequence of questions and answers that fixes a mistake of a use when the knowledge tree of the program is as in first page of the project:

Program: Are you a mammal? [Y or N or U]
User: Y
Program: Are you bigger than a cat? [Y or N or U]
User: Y
Program: My guess is Kangaroo. Am I right? [Y or N or U]
User: U
Program: Is the question "Are you bigger than a cat?" where you made a mistake? [Y or N or U]
User: N
Program: Is the question " Are you a mammal? " where you made a mistake? [Y or N or U]
User: Y
Program: Do you live underwater? [Y or N or U]
...

Part VI.

- (1) How much time did you spend for Project 2?
- (2) How much time did you spend on Part I in minutes?
- (3) How much time did you spend on Part II in minutes?
- (4) How much time did you spend on Part III in minutes?

(Note: The following equation must hold:

Answer to (1) = Answer to (2) + Answer(3) + Answer(4).)