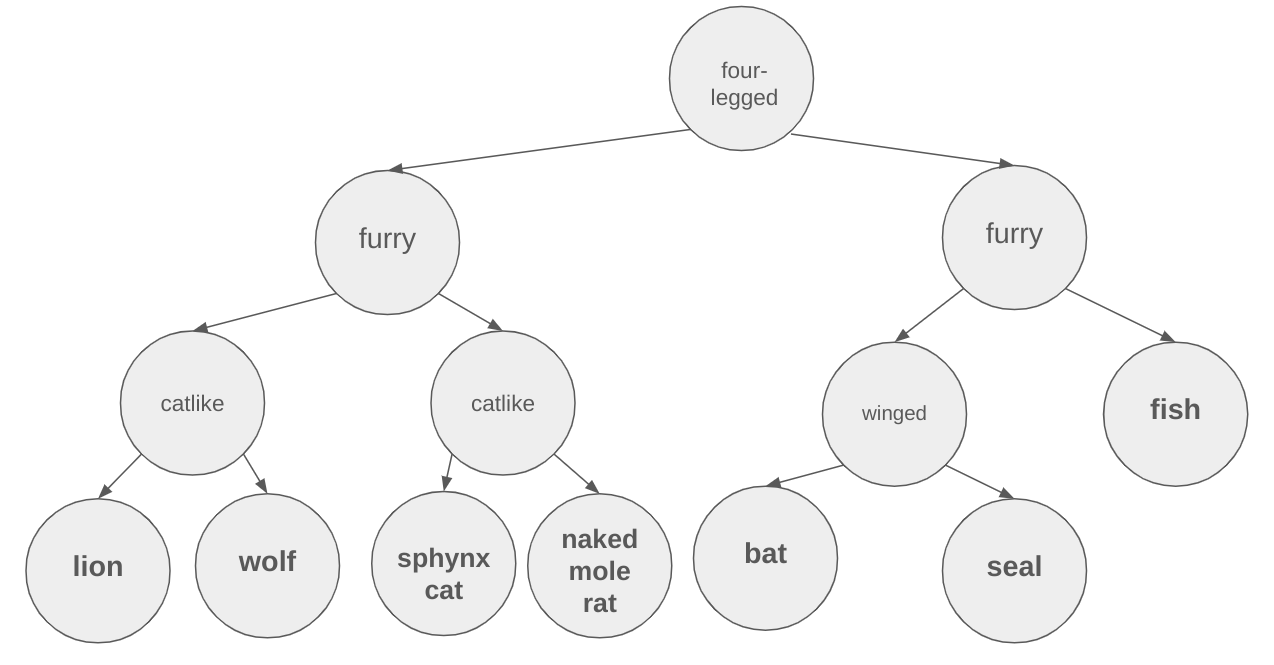
**Unit 3 Project - 20 questions**

# Instructions

In the game of 20 Questions, one person thinks of an object (often an animal), and another player asks a series of yes-or-no questions until they can identify the object. In this assignment, you will write a program that can play 20 Questions to guess what you're thinking of.

You will begin by generating the data that powers the game, which involves labeling various objects with binary classifications (a lion is four-legged, furry, and catlike, but naked mole rat is four-legged, not furry, and not catlike). It would be a lot of work to generate this data yourself, so you will lean on genAI.

Once you have your data saved to a file, you'll construct a binary tree representing the classifications of these objects. Maybe the root represents the question "Is it four-legged?" and any nodes in the left subtree are four-legged and any nodes in the right subtree are not four-legged. The left and right subtree might have questions in the same order, but they may not. Some nodes will contain specific guesses (e.g. lion or naked mole rat). Your whole tree might look like this:



When you construct your tree (and yes, you must construct a tree), you must construct it in a way that is **better than random**. To be more specific, you must **at least** select whatever question most evenly splits the remaining possible answers into two groups. Think binary search! This minimizes the height of the longest branch in your tree, and thus the worst case for your program's guessing strategy.

Your program also needs to work on arbitrary data, as long as it's formatted the same way as your file. That is, there might be a different number of rows, or even different category names, but your program still needs to work on that file.

Furthermore, your program should work on any input file, as long as it’s formatted the same way as your generated file. Avoid hardcoding specific question names / ordering or categories; the program must infer these details from the input itself.

The program should flow as follows:

* When the user runs the program, the program should build a decision tree by reading the raw data from the data file.
* Then, the program should run one round of the game:
  + The program should ask a series of yes / no questions to the user to determine what object they were thinking of.
    - Your program should gracefully handle unexpected input
  + If the program gets a definitive "yes" answer for a specific object, it should print a gracious victory message with the number of guesses it took.
  + If the program could not identify the object, it should congratulate the user on a game well played.
* Regardless of the outcome, the program should ask the user if they want to play again, and if so, begin another round of the game, without having to rebuild the entire tree.

A run of your game might look something like this:

| Is your animal four-legged?  > yes  Is your animal furry?  > yes  Is your animal catlike?  > no  Is your animal a wolf?  > yes  Good game! I won in 4 guesses. Would you like to play again?  > no  Goodbye! |
| --- |

### Deliverables:

* Checkpoint, due 3/30
* Code Implementation, due 4/8
* TA Meeting, by 4/14

### Evaluation Criteria:

Your submission will be graded according to the following rubric:

| Checkpoint | 10% |
| --- | --- |
| Decomposition | 15% |
| Implementation | 50% |
| TA Meeting | 25% |

## Teamwork and Roles:

This project is to be completed individually. You may discuss any aspects of the project with course staff. You may discuss the project at a high level with classmates, but you should not share code or answers to specific questions.

### Use of genAI:

* **Category 1 - Free use of AI tools**
  + You are encouraged to use generative AI to help with this project, especially to generate Python functions from your own descriptions. You can find more details below.

## Phase 1 (Checkpoint)

You will first need to generate the data to use in your 20 Questions game. Choose a broad category (animals, movies, cities, etc.). You should be able to imagine a dataset with *hundreds* of objects of the type you chose, labeled for at least 8 binary classifications each. You should also think of a single category (like one of the ones mentioned), rather than including all types of objects.

When I asked an LLM to give me list of 512 animals each labeled with 12 characteristics, which returned a response like this (with many more rows):

| **Animal** | **Mammal** | **Bird** | **Fish** | **Reptile** | **Amphibian** | **Insect** | **Domestic** | **Wild** | **Herbivore** | **Carnivore** | **Omnivore** | **Aquatic** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lion | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| Eagle | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Salmon | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

Eventually you will need the data in a format you can parse. You can continue prompting the LLM until it is in such a format, or you can take what you get initially and think about the code you will need to parse it.

For your checkpoint, you must turn in:

* A test dataset with 10 items and 3 binary classifications, in CSV or JSON format. This will be crucial to testing your program!
* Your full dataset, consisting of at least 100 items, labeled for at least 8 binary classifications, in CSV or JSON format
* The python code defining your TreeNode class
* A decomposition diagram
* Answers to the following questions:
  + How are you going to differentiate between items that have the same value for every label?
  + How will you pick the first question to ask? In each subtree, how will you pick the next question to ask?

Reach out to the course staff ASAP if you are having trouble generating data. We can provide you with a dataset so that you can continue working on the project.

## Phase 2: Implementation

Implement the game according to the description and all listed requirements. You will be graded on your design and the justification of your design. If you have questions, please reach out to your instructor or TA.