

CS 225 Final Project | Results

Read File

Using our modified data set, we were able to accurately read in information from a text file and construct a taxonomic tree. The first line of the text file, Animalia, was correctly assigned as the root of the tree, and it correctly had 33 children, all of which are phylums. The getline function was used to store each entry in the text file in an array of strings, which was then used to create the tree structure. Each entry was scanned for the brackets that contain the entry's level in the taxonomic tree, and it was assigned to its parent.

Find

We have two find methods. The first takes in a string, and returns the first node in the tree that contains the string. This first method is generally used within the class as part of encapsulation, as the following three methods all must first find nodes before they can execute their functions. The second also takes in a string, but returns the rest of the entry that the node refers to, for use outside of the class. The input and output are displayed below for a query of *Elephantulus edwardii*, the Cape elephant shrew. If the query is invalid, find simply returns "query not found".

```
[jackts2@linux-03 finalProject]$ ./treemaker
Input query for find: Elephantulus edwardii

Return value: Elephantulus edwardii (A. Smith, 1839) [species]

Input query for find: Primate

Return value: Primates Linnaeus, 1758 [order]
```

Find Most Connected

This method also has two versions, but the purposes are solely for encapsulation. The first is responsible for taking the input string and finding the starting node. It then calls the second method, which takes in a node as the subroot of a tree. The method then iterates through each node in the subtree, and returns the most connected node. The first method then accesses the entry the node corresponds to and returns the full string.

The image below displays the most connected node within the order Primates: Lepilemur. This genus consists of most Lemurs found on the island of Madagascar, which is known to be an incredibly diverse group of primates that are closely related, showcasing that lemurs are the most biologically diverse group of primates.

```
Input query for mostConnected: Primates  
Return value: Lepilemur I. Geoffroy Saint-Hilaire, 1851 [genus]
```

Find Distance

Find distance returns an integer that represents how many edges are between two queries. For example, in the image below two arguments were given: *Elephantulus edwardii*, the Cape elephant shrew; and *Homo sapiens*, humans. The elephant shrew and humans are only in the same order, Mammalia. This means that in order to travel from humans to the elephant shrew, one must pass through genus *Homo*, family Hominidae, and order Primates before arriving at Mammalia. This constitutes a distance of four. To then travel to the elephant shrew, one must pass through order Macroscelidea, family Macroscelididae, and genus *Elephantulus* before arriving at the species. Once more this is a distance of four. Adding it together, a total distance of 8 is produced.

```
Input first query for findDistance: Elephantulus edwardii  
Input second query for findDistance: Homo sapiens  
Return value: 8
```

Lowest Common Ancestor

This method returns the entry that serves as the last common ancestor node of two inputs. As seen above, to travel from *Homo sapiens* to *Elephantulus edwardii* one must travel to Mammalia and then back down. This is because Mammalia is the lowest common ancestor of those two nodes. This function then returns the entry of Mammalia.

```
Input first query for lowestCommonAncestor: Elephantulus edwardii  
Input second query for lowestCommonAncestor: Homo sapiens  
Return value: Mammalia Linnaeus, 1758 [class]
```