

## CONSTRUCTING AN EVOLUTIONARY TREE BY ENHANCING A GENERAL TREE ADT

[The Tree of Life Web Project \(ToL\)](#) is a collaborative effort of biologists and nature enthusiasts from around the world. The project provides information about biodiversity, the characteristics of different groups of organisms, and their evolutionary history. The aim is to create a web page for every species and group of organisms, living or extinct. These pages are linked hierarchically, forming a branching Evolutionary Tree (Phylogenetic Tree) that illustrates the genetic connections between all living things.



In this assignment, you are expected to develop a Java program to represent the ToL dataset with a General Tree ADT and provide some specified functionalities.

### 1. Dataset

ToL dataset has a tree structure and contains 35960 species and links between all the nodes. It is a branching diagram showing the evolutionary relationships between various biological species. Its organization consists of the following levels. The arrows point from more specific to more general.

life <- domain <- kingdom <- phylum <- class <- order <- family <- genus <- species

Key aspects of an Evolutionary Tree:

1. **Root:** This is the common ancestor of all the organisms represented in the tree.
2. **Branches:** These represent the evolutionary path from the common ancestor to the various descendant species or groups.
3. **Nodes:** Points where branches split, indicating a common ancestor that diverged into two or more lineages.
4. **Leaves (or Tips):** The end points of the tree, representing individual species or groups that exist today or existed in the past.

## 2024-2025 FALL

### CME 2201 - Assignment 2

You will be given two data files that contain the nodes and links.

#### 1.1 Species File (treeoflife\_nodes.csv)

node_id	Numeric identifier for the species in the tree
node_name	Name of the species or none if unknown
child_nodes	Number of child nodes
leaf_node	Whether or not the node is a leaf
tolorg_link	Whether or not there is a page on the website tolweb.org describing this species. <a href="http://tolweb.org/\${node_name}/\${node_id}">http://tolweb.org/\${node_name}/\${node_id}</a>
extinct	Whether the species is living or extinct 0 - living , 1 - extinct species
confidence	Confidence of placement in the tree structure. 0 - confident position , 1 - problematic position , 2 - unspecified position
phylesis	0 - monophyletic , 1 - uncertain monophyly , 2 - not monophyletic

#### 1.2 Tree Links File (treeoflife\_links.csv)

source_node_id	Ancestor or source node identifier
target_node_id	Descendant or target node identifier

## 2. Data Structure

In this assignment, you will develop your own Evolutionary Tree structure to represent LoT dataset in Java programming language. The Evolutionary Tree must have a general tree structure that contains nodes with zero or more children. You must use a Hashtable to access species records with their ids, rapidly. You should also keep references for parent nodes to follow the species' ancestor paths. You are free to use the built-in classes provided by the Java library including Hashtable, List, Stack, Queue etc. But you must implement the Tree classes. An illustration of the Evolutionary Tree structure is provided in Figure 1.

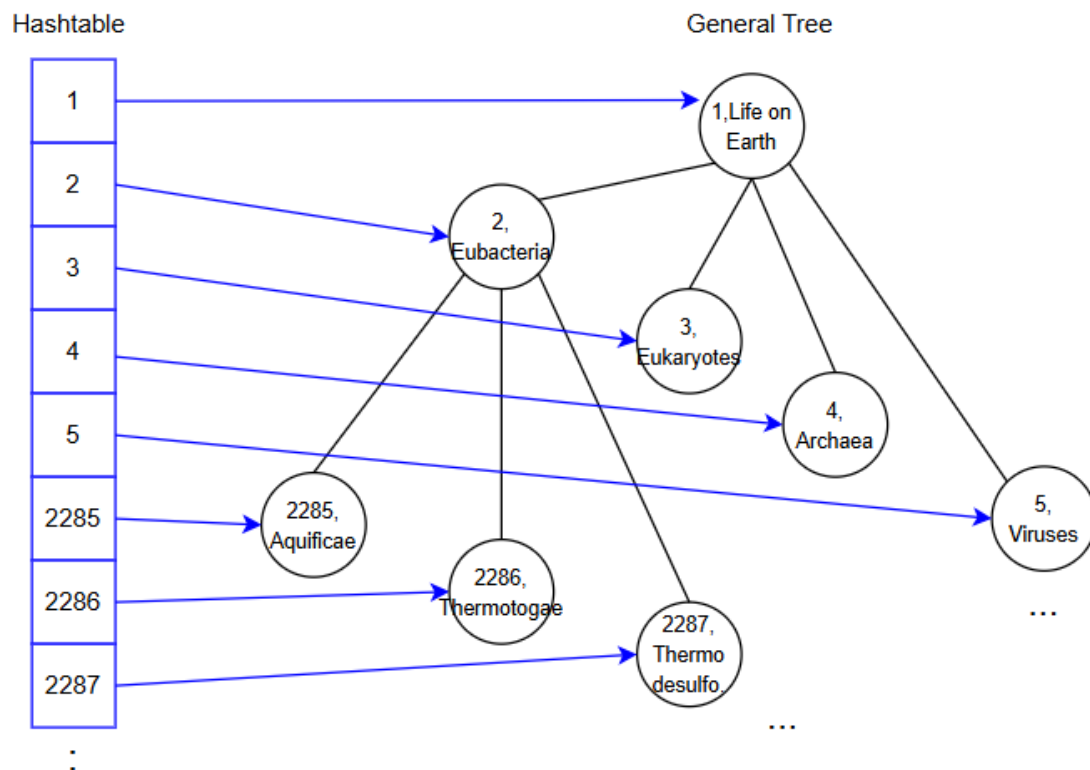


Figure 1. An illustration of the Evolutionary Tree

#### 4. Functionality

A Program Menu should be provided for ease of use. The menu should include:

- 4.1. Load dataset
- 4.2. Search for species.

Input: 16421	Output: Id: 16421 Name: Homo sapiens Child count: 0 Leaf node: yes Link: <a href="http://tolweb.org/Homo_sapiens/16421">http://tolweb.org/Homo_sapiens/16421</a> Extinct: no Confidence: no Phylaxis: no
--------------	--

- 4.3. Traverse the tree  
Traverse the tree in pre-order and save the traversal order into the file pre-order.txt.

## 2024-2025 FALL

### CME 2201 - Assignment 2

4.4. Print the subtree of a given species in pre-order.

Input: 16299	Output: 16299-Hominidae (+) -16410-Pongo (+) ... -----16418-Homo (+) -----16421-Homo sapiens (+) -----16422-Homo erectus (-) -----16423-Homo ergaster (-) -----16424-Homo rudolfensis (-) -----16425-Homo habilis (-) --16419-Gorilla (+) ---65355-Gorilla beringei (+) ---65356-Gorilla gorilla (+)
-----------------	--

4.5. Print the ancestor path of the given species.

Input: 16421	Output: 1-Life on Earth (+) -3-Eukaryotes (+) --2372-Opisthokonts (+) ---2373-none (+) ----2374-Animals (+) . . . -----16418-Homo (+) -----16421-Homo sapiens (+)
-----------------	---

4.6. Find the most recent common ancestor of the given two species.

Input: 16421 16954	Output: The most recent common ancestor of 16421-Homo sapiens and 16954-Microhylidae is 14987-Tetrapoda.
--------------------	--

4.7. Calculate the height, degree and breadth of the tree.

For a given node, the degree is its number of children. A leaf, by definition, has a degree zero. The degree of a tree is the maximum degree of a node in the tree. The breadth means the number of leaf nodes in the tree.

4.8. Print the longest evolutionary path\paths.

## 5. Project Groups

You will develop the project individually!!!

## 6. Due date

**16.01.2025 Thursday 23:55.** Late submissions are not allowed. The due date will not be extended.

## 7. Requirements

- Usage of Java programming language and Abstract Data Types (ADT) are required.
- Object Oriented Programming (OOP) principles must be applied.
- Exception handling must be used when it is needed.

## 8. Submission

You must prepare a project report describing your data structure, java code and output. Your report should contain a cover page.

You must upload your all 'java' files and report as an archive file (.zip or .rar). Your archived file should be named as 'studentnumber\_name\_surname.rar.zip' (e.g., 2007510011\_Ali\_Yılmaz.rar) and should be uploaded via **online.deu.edu.tr**.

## 9. Code Control

The project will be controlled in laboratory sessions on 17 January 2025. A schedule will be announced before the code control date. You must be in the laboratory on time. You will have 10 minutes to show your assignment. Please do not forget to bring your laptops while coming to the assignment control!

## 10. Plagiarism Control

The submissions will be checked for code similarity. Copy assignments will be graded as zero.

## 11. Grading Policy

Job	Percentage
Usage of ADT, OOP and Try-Catch	%30
Evolutionary Tree implementation	%50
Program menu and functionality	%20

## References

Maddison, D. R. and K.-S. Schulz (eds.) 2007. The Tree of Life Web Project.  
<http://tolweb.org>

GOOD LUCK!