

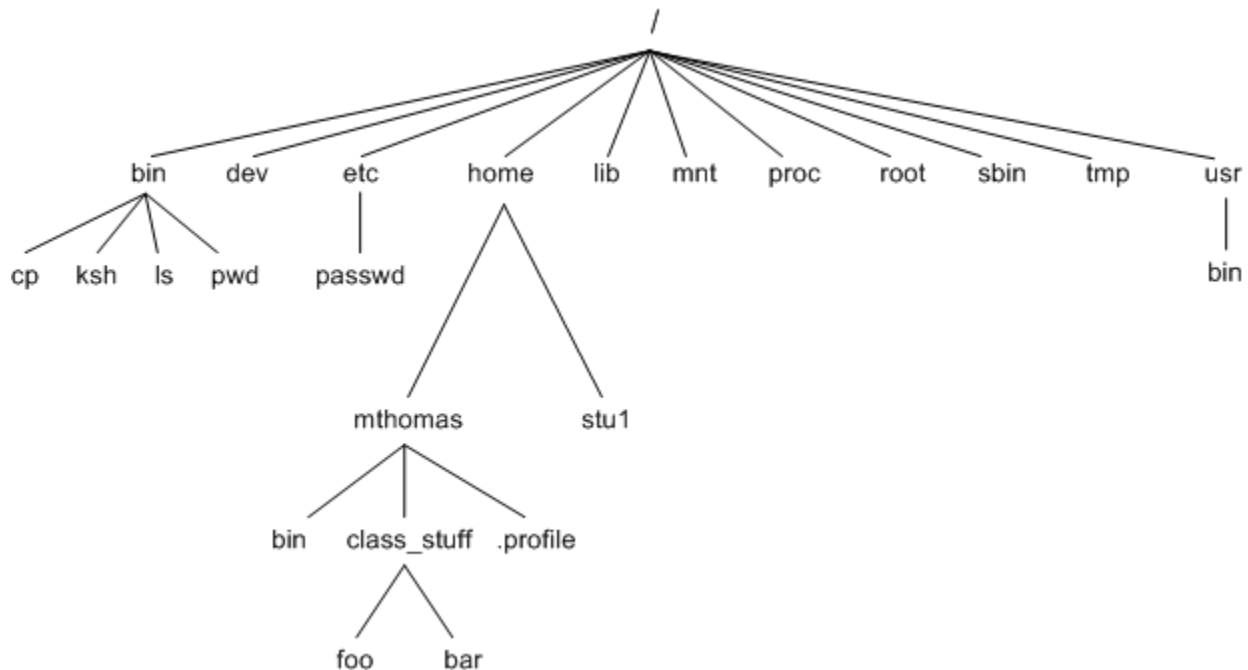
**Lab 11 – 9<sup>th</sup> April**

**Topics – Lowest Common Ancestor (LCA)**

**Objective**

In this exercise we will implement an algorithm (as well as the data structure) to find the Lowest Common Ancestor (LCA) of a given n-ary rooted tree. The LCA problem is defined as follows: Given a rooted tree **T** and two nodes **u** and **v**, find the furthest node from the root that is an ancestor for both **u** and **v**. Here is an example.

In the first part of this exercise, we will construct a tree data structure from input data. To realize this, we take an example of UNIX file system in which the files are organized into a multi-leveled hierarchy called a directory tree. At the very top of the file system is single directory called "root" which is represented by a / (slash). All other files are "descendents" of root.



After constructing the tree based on the input data, you have to implement an algorithm to find out the LCA of two different nodes in the tree. In the above example "home" is the lowest common directory for the directories bin and stu1.

**Input format**

Each line will start with a one of the following key values (1, 2, or -1). Each key corresponds to a function and has a pattern. Implement following function according to given pattern and write a driver function which can call the functions according to given key value.

Key	Function to call	Input Format	Description
1	<i>readData</i>	1 N path <sub>1</sub> path <sub>2</sub> ..... path <sub>N</sub>	Read next N lines having a path in each line. "path" will be a system path of a file or directory (containing letters, numbers, spaces and arbitrary number of slashes). Keep inserting each path in a tree as mentioned above.

2	LCA	2 path <sub>1</sub> path <sub>2</sub>	search for “Lowest Common Ancestor” of both paths in the tree. If only root is LCA (i.e. nothing matches), then print 0, otherwise print the child index (i.e. which number of child the common child is, of its parent, starting from 1). For each common level, child index should be printed, separated by space.
-1			stop the program.

**Test Case 1:**

Input	Output
1 10 /bin /dev/hd/nano /etc /home /lib /usr/nano /var /dev/fd /dev/hd/dmuser /dev/tty /usr/saiyedul /user/jagat 2 /dev/hd/dmuser /dev/hd/nano -1	P Q where “dev” is P <sup>th</sup> child of “/” and “hd” is Q <sup>th</sup> child of “dev”.