

## Lab Projects

### Data Structures & Algorithm Lab (CS392)

#### 1. Decision Tree:

Explore Decision tree classifier and implement it. Use a suitable tree data structure to implement the classifier. Use the following data to train and test the program;

<i>RID</i>	<i>age</i>	<i>income</i>	<i>student</i>	<i>credit rating</i>	<i>Class: buys computer</i>
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle	medium	no	excellent	yes
13	middle	high	yes	fair	yes
14	senior	medium	no	excellent	no

#### 2. Trie:

Trie is a kind of search tree used to store dynamic sets and the keys in each node are usually strings. One of the applications of trie is predictive text or autocomplete feature in word processors, web browsers etc. Implement a trie and use it to produce a probable list of words that a user can write using first 3 to 5 letters typed by the user. Use a dictionary having reasonable number of words to test the application.

#### 3. File management system:

The system should follow the working principle of 'ls' command in Linux Operating system. It should be able to show the files, folders under the current directory. The user should be able to add file, folders under specific folder. If the folder under which the user want to add the files do not exist, it should be created first and then the files/folders should be added. Use tree data structure to implement the system. Please note that this files or folders mentioned above need not to be created physically. Storing the names would be sufficient.

#### 4. Routing:

A network consists of several computing devices. There could be many different ways through which a message from one computer can reach another computer. The network can be thought of as a graph. The weights of the graph indicate the cost associated with different links (connection between nodes). Given two different nodes, the system should first check if a path is available to reach the destination node or not. If the path is available then find the path having least cost. Also implement a buffer to store last 10 request results, if in future the same route is searched, it can be served faster.

#### 5. Tic-Tac-Toe:

Implement an application to play tic-tac-toe game against computer. Use tree data structure to store the game states.

#### 6. Spell Checker:

Spell checker is used to check if a user typed word is correctly spelled using a dictionary of correct words. Word processors (e.g. MS-Word), electronic mail, web search engines and many other applications use spell checkers. The application should be able to check if a user typed word is correct or not based on dictionary look up. Also the application should suggest the nearest word from the dictionary. To test the application, use a dictionary having reasonable number of words.

**Project Allotment List**

<b>Project No.</b>	<b>Group no.</b>
<b>1</b>	<b>1 – 8</b>
<b>2</b>	<b>8 – 15</b>
<b>3</b>	<b>16 – 23</b>
<b>4</b>	<b>24 – 31</b>
<b>5</b>	<b>32 – 39</b>
<b>6</b>	<b>40 – 47</b>