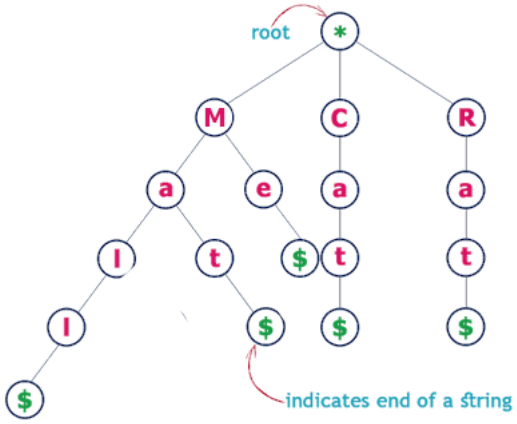
An-Najah National University

Computer Engineering Department

Algorithms and Computational Complexity

Assignment #2 String Retrieval Using Trees

In this assignment you will implement a special data structures for storing, retrieving & searching strings in an efficient way. This data structure is a Tree where each node represents a character. Each path in the tree starting from the root and ending in one of the leaves represents a certain string. The leaves contain a special character '$' which indicates the end of the string. The following figure is an example of this data structure.



Your task is to build this data structure by implementing the following operations:

- Insert (char *s). This operation inserts the string **s** in the tree.
- Delete (char *s). This operation deletes the string **s** from the tree.
- Search (char *s). This operation prints Yes or No based on whether the string **s** exists in the tree or not.
- StartsWith(char *s). This operation prints all strings that start with string **s**. Strings are sorted then printed.
- Longest(). This operation prints the longest string stored in the tree (assume there is only one such string).

For example to build the tree shown in the figure above, you would first create an empty tree then call the following functions:

```
Insert("Mall");
Insert("Me");
Insert("Mat");
Insert("Cat");
Insert("Rat");
```

Also if you call the following function:

```
StartsWith("Ma");
```

Then the following is printed:

```
Mall
Mat
```

If you call the following function:

```
Longest();
```

Then the following is printed:

```
Mall
```

To test your program you should expect a transaction file. The first line contains the number of commands that will be executed on the tree. After that each line in the file represents a command that either: inserts, deletes, searches, etc. entries based on the key. Here is an example of a test case file.

Dr. Samer Arandi

Good Luck ☺