Indian Institute of Technology Roorkee
Department of Computer Science and Engineering

CSN-261: Data Structures Laboratory (Autumn 2019-2020)

Lab Assignment – 4

Name: Hemil Sanjaybhai Panchiwala

Enrollment No.: 18114031

Branch: Computer Science & Engineering

Sub Batch: O2

Problem Statement 1

Create a dictionary using Trie data structure (without using STL) having words and their meanings. You need to read the words and their respective meanings from a CSV file (uploaded in Piazza, named as TrieInput.csv), where 1st column is for words and 2nd column shows its meaning.Given a word you have to print its meaning. If no such word is found in the dictionary, then print “Invalid word”. Create a GUI using Qt library to accept a word in a text box and display the meaning in an anotherbox, as shown in the Figure 1.

Also,create an installer of your program for Windows OS. You can use the software like InstallSimple or InstallShield or WIX or NSIS to do so.

Data Structures used :-

Tree

Algorithm

- **I have created a trie using tree data structure in which I have implemented different methods like insert, search for implementation of dictionary.**

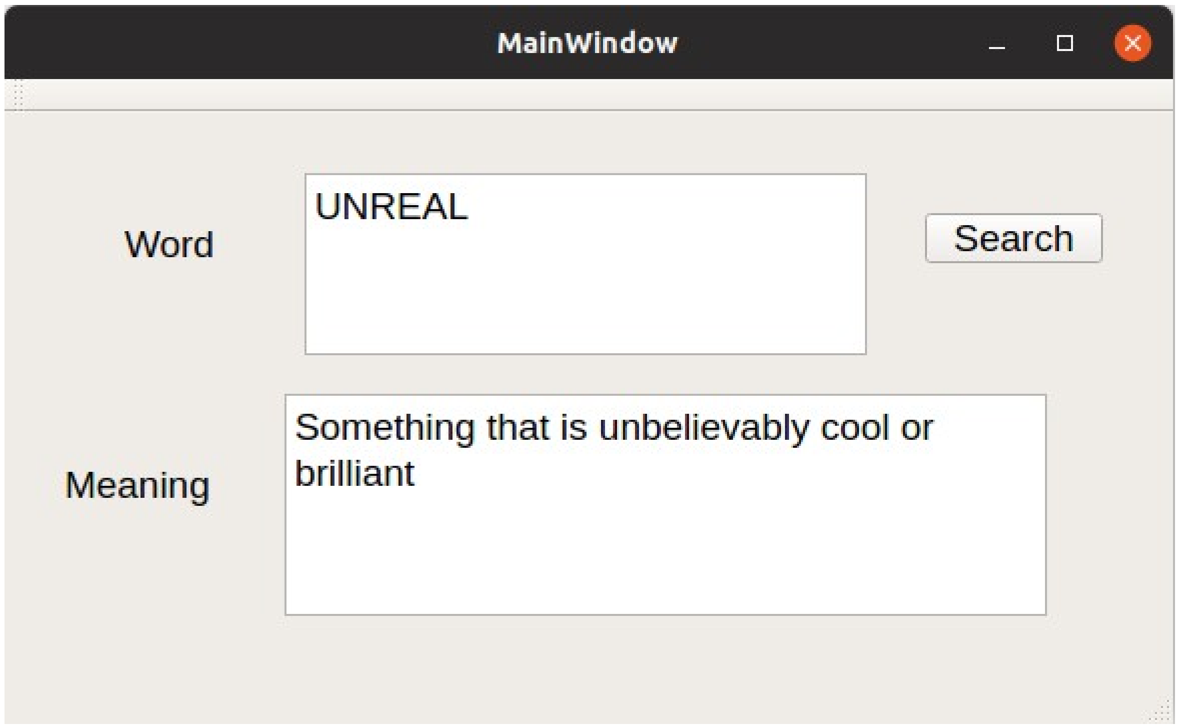
➔ Insertion

- **Set a current node as a root node**
- **Set the current letter as the first letter of the word**
- **If the current node has already an existing reference to the current letter, then set current node to that referenced node. Otherwise, create a new node, set the letter equal to the current letter, and also initialize current node to this new node**
- **Repeat step 3 until the string is traversed**

➔ Search

- **Get children of the root**
- **Iterate through each character of the *String***
- **Check whether that character is already a part of a sub-trie. If it isn't present anywhere in the trie, then stop the search and return "No word found!!!"**
- **Repeat the second and the third step until there isn't any character left in the String. If the end of the String is reached, return the meaning of the string**

Snapshots



Word: UNREAL

Meaning: Something that is unbelievably cool or brilliant

Problem Statement 2

Implement N Queens problem to show all the possible combinations inN x Nbinary matrix and to display the total number of such combinations at the end, where 1 represents the position of N queens in theN x N matrix and remaining cells are represented by 0.

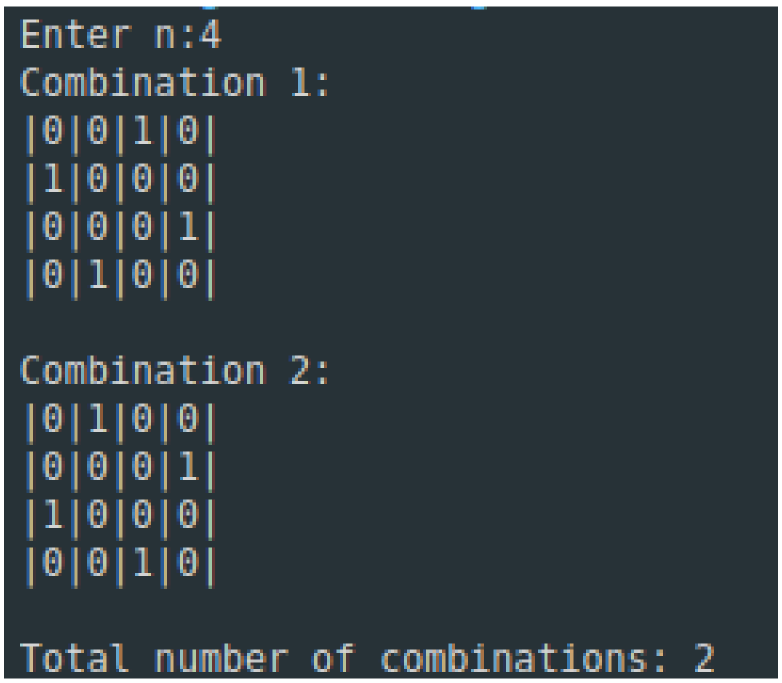
Data Structures used :-

Array

Algorithm

```
    1. Place the queens column wise, start from the left
most column
    2. If all queens are placed.
        print the solution matrix.
    3. Else
        1. Try all the rows in the current column.
        2. Check if queen can be placed here safely if yes
        mark the current cell in solution matrix as 1 and
        try to solve the rest of the problem recursively.
        3. If placing the queen in above step leads to the
        solution return true.
        4. If placing the queen in above step does not lead
        to the solution , BACKTRACK, mark the current
        cell in solution matrix as 0 and return false.
    3. If all the rows are tried and nothing worked, return
    false and print NO SOLUTION.
```

Snapshots



```
Enter n:4
Combination 1:
|0|0|1|0|
|1|0|0|0|
|0|0|0|1|
|0|1|0|0|

Combination 2:
|0|1|0|0|
|0|0|0|1|
|1|0|0|0|
|0|0|1|0|

Total number of combinations: 2
```

Time taken: 0.000234 s

Problem Statement 3

Given an integer array having N number of elements, write a C++ programusing hash map (using STL) to find the length of the largest subarray from the given input array, where thesummation of the elements of the subarray is equal to n. In the output, if any solution exists then print the starting and ending index (with respect to given input array) of the largest subarray and also print its length. Otherwise, print “Not Found”, as described in the following output.

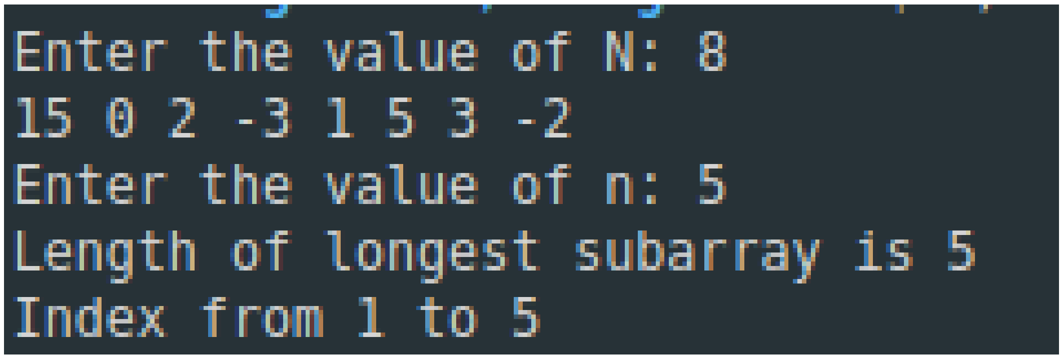
Data Structures used :-

Array, HashMap

Algorithm

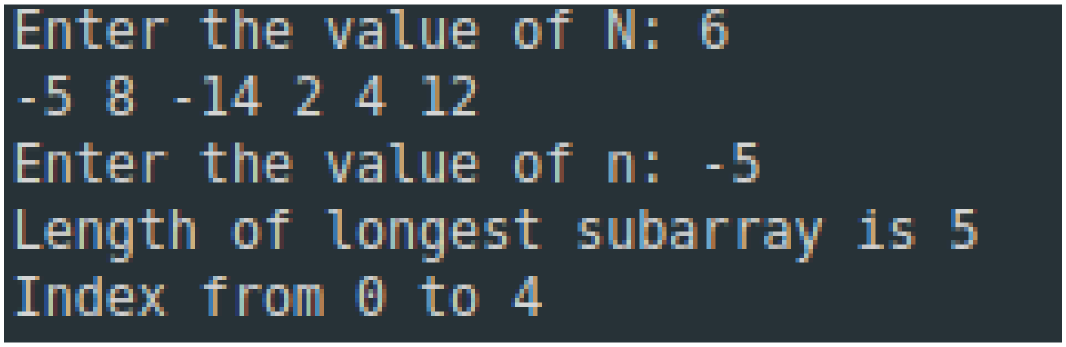
```
1.Initialize sum = 0 and maxLen = 0.
2.Create a hash table having of sum and index.
3.Loop from i = 0 to n-1
    1.Add arr[i] to sum.
    2.If sum == k, update maxLen = i+1.
    3.Check whether sum is present in the hash table or
      not. If not present, then add it to the hash table.
    4.Check if (sum-k) is present in the hash table or
      not. If present, then obtain index of (sum-k) from
      the hash table as index. Now check if maxLen <
      (i-index), then update maxLen = (i-index).
4.Return maxLen.
```

Snapshots



```
Enter the value of N: 8
15 0 2 -3 1 5 3 -2
Enter the value of n: 5
Length of longest subarray is 5
Index from 1 to 5
```

Time taken: 0.000351 s



```
Enter the value of N: 6
-5 8 -14 2 4 12
Enter the value of n: -5
Length of longest subarray is 5
Index from 0 to 4
```

Time taken: 0.000219 s