CSE- 3004 LAB- Assignment

Academic year: 2020-2021 **Semester:** WIN

Faculty Name: Dr. D Sumathi Mam **Date:** 17 /11/2021

Student name: M.Taran **Reg. no.:** 19BCE7346

Implement any of the following methods for your choice of problem. Methods are dynamic programming, Backtracking and Branch and bound

The Word Break Problem Using Backtracking

Objective :

For a Given string and a dictionary of words,we need to find out if the input string can be broken into a space-separated sequence of one or more dictionary words.

So, here we need to Navigate the given input string. And Take a blank string and keep adding one character at a time to it. and Keep checking if the word exists in the dictionary.

So, If a word exists in the dictionary then add that word to the answer string and make a recursive call to the rest of the string. If any of the recursive calls returns false then backtrack and remove the word from the answer string and again keep adding the characters to the string.

If all the recursive calls return true that means the string has been broken successfully.

Code :

```
import java.util.HashSet;
```

```
public class WordBreakRecursion {
    public void wordBreak(String s, HashSet<String> hs) {
        if (find(s, hs, "")) {
        } else {
          System.out.println("Can't Break this sentence");
        }
      }

      public boolean find(String s, HashSet<String> dict, String answer) {

        if (s.length() == 0) {
          System.out.println(answer);
          return true;
        } else {
          int index = 0;
          String word = "";
          while (index < s.length()) {
            word += s.charAt(index);
            if (dict.contains(word)) {

              if (find(s.substring(index + 1), dict, answer + word + " "))
{
                return true;
              } else {
                      index++;
              }
            } else {
              index++;
            }
          }
          return false;
        }
      }
      public static void main(String[] args) {
        HashSet<String> hs = new HashSet<String>();
        hs.add("hello");
        hs.add("iam");
        hs.add("taran");
        hs.add("from");
        hs.add("the");
        hs.add("all");
        hs.add("this");
```

```
        hs.add("is");
        hs.add("and");
        hs.add("vit");
        String s = "helloallthisistaranandiamfromvit";

        WordBreakRecursion ws = new WordBreakRecursion();
        ws.wordBreak(s, hs);
      }
}
```

OUTPUT:

Result

compiled and executed in 0.958 sec(s)

```
hello all this is taran and iam from vit
```