

4.9 WORD BREAK PROBLEM WITH PREDEFINED DICTIONARY

Question:

Given an input string and a dictionary of words, find out if the input string can be segmented into a space-separated sequence of dictionary words. Consider the following dictionary { i, like, sam, sung, samsung, mobile, ice, cream, icecream, man, go, mango }.

AIM

To implement a Python program that checks whether a given string can be segmented into valid dictionary words using recursion and memorization.

ALGORITHM

1. Define a set dictionary containing all valid words.
2. Use a recursive function `can_segment(s)` to check if the string `s` can be broken into valid words.
3. For each prefix of `s`, check if it exists in the dictionary.
4. If it does, recursively check the remaining suffix.
5. Use memorization to avoid redundant computations.
6. Return True if any valid segmentation exists.

PROGRAM

```
def can_segment(s, wordDict):
    dp = [False] * (len(s)+1)
    dp[0] = True
    for i in range(1, len(s)+1):
        for word in wordDict:
            if i >= len(word) and s[i-len(word):i] == word and dp[i-len(word)]:
                dp[i] = True
    return "Yes" if dp[-1] else "No"

dictionary = ["i", "like", "sam", "sung", "samsung", "mobile", "ice", "cream", "icecream", "man", "go", "mango"]
s = input("Enter the string: ")
print("Can the string be segmented?", can_segment(s, dictionary))
```

Input:

Enter the string: ilikeicecream

Enter the string: ilikesamsung

Output:

```
>>> Enter the string: ilikeicecream
      Can the string be segmented? No
=====
>>> Enter the string: ilikesamsung
      Can the string be segmented? Yes
>>> |
```

RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity: $O(n^2)$
- Space Complexity: $O(n)$