2707. Extra Characters in a String

You are given a **0-indexed** string `s` and a dictionary of words `dictionary`. You have to break `s` into one or more **non-overlapping** substrings such that each substring is present in `dictionary`. There may be some **extra characters** in `s` which are not present in any of the substrings.

Return *the* ***minimum*** *number of extra characters left over if you break up* `s` *optimally.*

Example 1:

```
Input: s = "leetscode", dictionary = ["leet","code","leetcode"]
Output: 1
Explanation: We can break s in two substrings: "leet" from index 0 to 3 and "code"
from index 5 to 8. There is only 1 unused character (at index 4), so we return 1.
```

Example 2:

```
Input: s = "sayhelloworld", dictionary = ["hello","world"]
Output: 3
Explanation: We can break s in two substrings: "hello" from index 3 to 7 and
"world" from index 8 to 12. The characters at indices 0, 1, 2 are not used in any
substring and thus are considered as extra characters. Hence, we return 3.
```

Constraints:

- `1 <= s.length <= 50`
- `1 <= dictionary.length <= 50`
- `1 <= dictionary[i].length <= 50`
- `dictionary[i]` and `s` consists of only lowercase English letters
- `dictionary` contains distinct words

2707. Extra Characters in a String

```
/*
   Recusrion + DP: memoization
   Time complexity: O(n^2 * m)
   Space complexity: O(n)
   n: size of the given string s
   m: size of the given dictionary
*/
class Solution {
public:
   int minExtraChar(std::string s, std::vector<std::string>& dictionary) {
      int n=s.size();
      std::unordered_map<std::string,int>memo;

      auto solve=[&](std::string s,auto& self)->int{
         if(memo.find(s)!=memo.end()) return memo[s];
         if(s=="") return 0;

         int ans=INT_MAX;
         for(auto& w: dictionary){
            // if w is a prefix of s
            if(s.substr(0,w.size())==w){
                std::string suff = s.substr(w.size());
                  ans=std::min(ans,self(suff,self));
            }
         }

         // Skip current character
         ans=std::min(ans,1+self(s.substr(1,s.size()-1),self));

         return memo[s]=ans;
      };

      return solve(s,solve);
   }
};
```