Problem # 524 : Longest Word in the Dictionary Through Deleting

<https://leetcode.com/problems/longest-word-in-dictionary-through-deleting/>

My Solution:

Runtime beats 62.13%.

1. Sort the dictionary in reverse order of length of the words in the dictionary and lexicographic order using key and lambda expression.

2. Iterate through the words in the dictionary. If the word is a subsequence of s, then return the word. If no word is found, then return an empty string.

3. For checking whether word is a subsequence of s, use the method is\_subsequence. Iterate through the letters of s and word and if all the letters in the word are present in s in the same order, return True. Otherwise return False.

class Solution:

def is\_subsequence(self, word, s):

'''

Return a boolean if word a subsequence of s

:param s: string

:param word: word

:return: boolean (True or False)

'''

n = len(s)

m = len(word)

i = 0

j = 0

while i < n and j < m:

if s[i]== word[j]:

j += 1

i += 1

return j == m

def findLongestWord(self, s: str, d: List[str]) -> str:

d = sorted(d, key = lambda x: (-len(x), x))

for word in d:

if self.is\_subsequence(word, s):

return word

return ""