Problem List

Run Submit

Description | Editorial | Solutions | Submissions

1065. Index Pairs of a String

Premium

Solved

Easy | Topics | Companies | Hint

Given a string `text` and an array of strings `words`, return *an array of all index pairs* `[i, j]` *so that the substring* `text[i...j]` *is in* `words`.

Return the pairs `[i, j]` in sorted order (i.e., sort them by their first coordinate, and in case of ties sort them by their second coordinate).

Example 1:

```
Input: text = "thestoryofleetcodeandme", words = ["story","fleet","leetcode"]
Output: [[3,7],[9,13],[10,17]]
```

Example 2:

```
Input: text = "ababa", words = ["aba","ab"]
Output: [[0,1],[0,2],[2,3],[2,4]]
Explanation: Notice that matches can overlap, see "aba" is found in [0,2] and [2,4].
```

Constraints:

- `1 <= text.length <= 100`
- `1 <= words.length <= 20`
- `1 <= words[i].length <= 50`
- `text` and `words[i]` consist of lowercase English letters.
- All the strings of `words` are **unique**.

Seen this question in a real interview before? 1/5

Yes No

Accepted 27.6K | Submissions 41K | Acceptance Rate 67.3%

Topics

Companies

Hint 1

Discussion (4)

375 4

Code

Python3 • Auto

```python
class Solution:
    def indexPairs(self, text: str, words: List[str]) -> List[List[int]]:

        ans = []

        for each in words:
            length = len(each)
            for i in range(0, len(text)-length+1):
                if text[i:i+length] == each:
                    ans.append([i, i+length-1])

        return sorted(ans)
```

Saved Ln 7, Col 31

Testcase | Test Result

Accepted Runtime: 69 ms

Case 1 Case 2

Input

text =
"thestoryofleetcodeandme"

words =
["story","fleet","leetcode"]

Stdout

```
thest thestoryofleetcodeandme
hesto thestoryofleetcodeandme
estor thestoryofleetcodeandme
story thestoryofleetcodeandme
```