

1181. Before and After Puzzle Premium

Solved 

Medium  Topics  Companies  Hint

Given a list of `phrases`, generate a list of Before and After puzzles.

A *phrase* is a string that consists of lowercase English letters and spaces only. No space appears in the start or the end of a phrase. There are no consecutive spaces in a phrase.

Before and After puzzles are phrases that are formed by merging two phrases where the **last word of the first phrase** is the same as the **first word of the second phrase**.

Return the Before and After puzzles that can be formed by every two phrases `phrases[i]` and `phrases[j]` where `i != j`. Note that the order of matching two phrases matters, we want to consider both orders.

You should return a list of **distinct** strings **sorted lexicographically**.

only two phrases are combined, no third one
eg. `[[a..b], [b..c], [c..d]]`
res will be:
`[[a..b..c], [b..c..d]]`

each phrase could be used multiple times

Example 1:

Input: `phrases = ["writing code", "code rocks"]`
Output: `["writing code rocks"]`

Example 2:

Input: `phrases = ["mission statement", "a quick bite to eat", "a chip off the old block", "chocolate bar", "mission impossible", "a man on a mission", "block party", "eat my words", "bar of soap"]`
Output: `["a chip off the old block party", "a man on a mission impossible", "a man on a mission statement", "a quick bite to eat my words", "chocolate bar of soap"]`

```
1 class Solution:
2     def beforeAndAfterPuzzles(self, phrases: List[str]) -> List[str]:
3         temp_dict = []
4         n = len(phrases)
5
6         for i, phrase in enumerate(phrases):
7             words = phrase.split()
8             head, tail = words[0], words[-1]
9             temp_dict.append([head, tail, i])
10
11         m = set()
12
13         for i in range(n):
14             for j in range(n):
15                 if i == j:
16                     continue
17                 if temp_dict[i][0] == temp_dict[j][1]:
18                     combine = phrases[j] + phrases[i][len(temp_dict[i][0]):]
19                     m.add(combine)
20
21         return sorted(list(m))
22
```

in fact, no need to use enum,
i is not important