1408. String Matching in an Array

Given an array of string words, return all strings in words that is a ***substring*** of another word. You can return the answer in **any order**.

A **substring** is a contiguous sequence of characters within a string

**Example 1:**

**Input:** words = ["mass","as","hero","superhero"]

**Output:** ["as","hero"]

**Explanation:** "as" is substring of "mass" and "hero" is substring of "superhero".

["hero","as"] is also a valid answer.

**Example 2:**

**Input:** words = ["leetcode","et","code"]

**Output:** ["et","code"]

**Explanation:** "et", "code" are substring of "leetcode".

**Example 3:**

**Input:** words = ["blue","green","bu"]

**Output:** []

**Explanation:** No string of words is substring of another string.

**Constraints:**

* 1 <= words.length <= 100
* 1 <= words[i].length <= 30
* words[i] contains only lowercase English letters.
* All the strings of words are **unique**.

class Solution {

    public List<String> stringMatching(String[] words) {

        List<String> inAnotherWordList = new ArrayList();

        for (int w = 0; w < words.length; w++) {

            // check if words[w] falls into any word

            String currentWord = words[w];

            int currentWordLength = currentWord.length();

            for (int i = 0; i < words.length; i++) {

                if (i == w) continue;

                String candidateWord = words[i];

                if (candidateWord.length() < currentWordLength) continue;

                if (candidateWord.indexOf(currentWord) != -1) {

                    // found a match

                    inAnotherWordList.add(currentWord);

                    break;

                }

            }

        }

        return inAnotherWordList;

    }

}

A screenshot of a phone

Description automatically generated