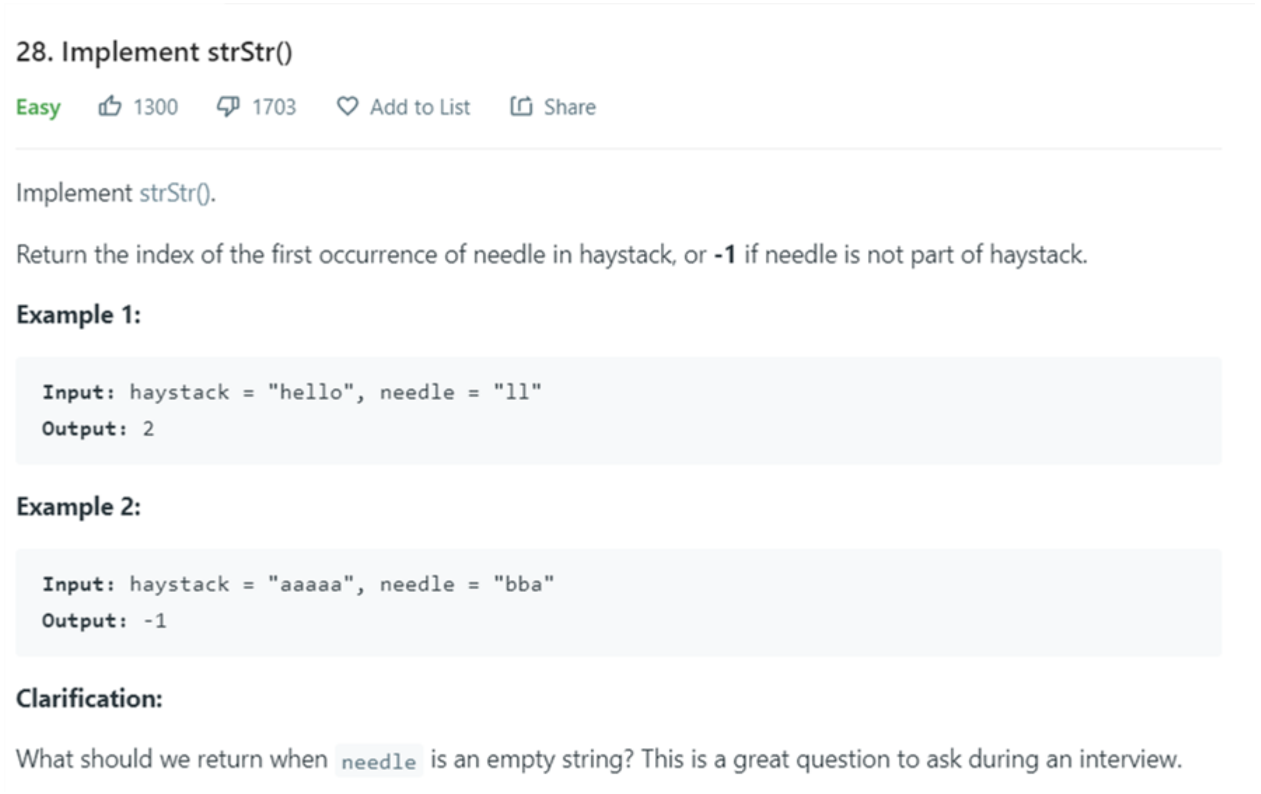
LeetCamp Week 2

Srishti Kumar
Nikhil Naru

UMPIRE Method (Codepath)

1. Understand
    a. Do a dry run through given example with interviewer
    b. Come up with edge cases
2. Match
    a. Have I already seen a version of this problem before?
3. Pseudocode (Plan)
4. Implement (will be very easy if you pseudocode first!)
5. Reflect (Test & Verify)
    a. Test your edge cases
6. Evaluate performance (Big-O Notation)

Question 1 - strStr()



28. Implement strStr()

Easy 1300 1703 Add to List Share

Implement strStr().

Return the index of the first occurrence of needle in haystack, or **-1** if needle is not part of haystack.

Example 1:

```
Input: haystack = "hello", needle = "ll"
Output: 2
```

Example 2:

```
Input: haystack = "aaaaa", needle = "bba"
Output: -1
```

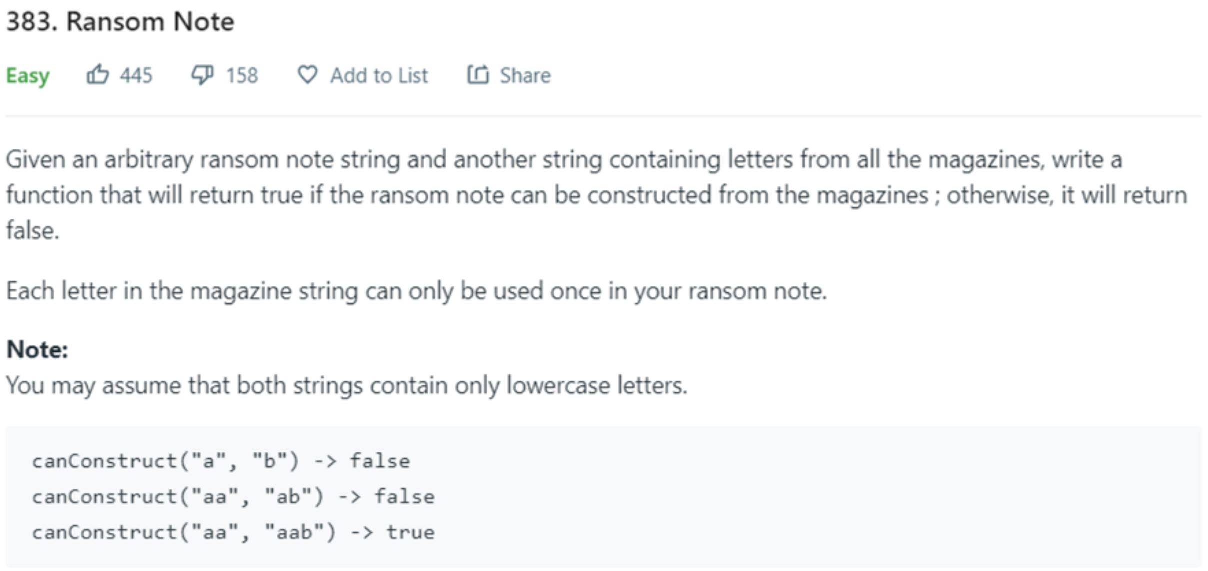
Clarification:

What should we return when `needle` is an empty string? This is a great question to ask during an interview.

Implement strStr()

- Brute force
  - Try all positions in string (2 for-loops)
  - O(N(N-L)) = O(N^2) = Quadratic Time
- 2-pointer approach
  - Better version of brute force
  - Best time complexity = O(N) = Linear
  - Worst time complexity = O(N(N-L)) = O(N^2) = Quadratic Time
- Rabin-Karp Algorithm (not discussed here - look it up if curious!)

Question 2 - Ransom Note



383. Ransom Note

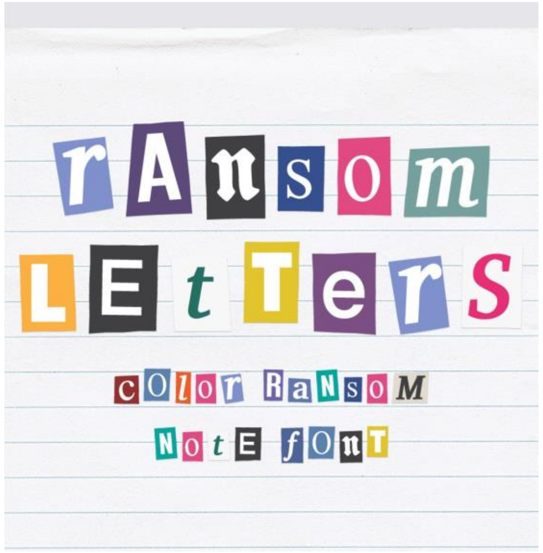
Easy 445 158 Add to List Share

Given an arbitrary ransom note string and another string containing letters from all the magazines, write a function that will return true if the ransom note can be constructed from the magazines ; otherwise, it will return false.

Each letter in the magazine string can only be used once in your ransom note.

Note:
You may assume that both strings contain only lowercase letters.

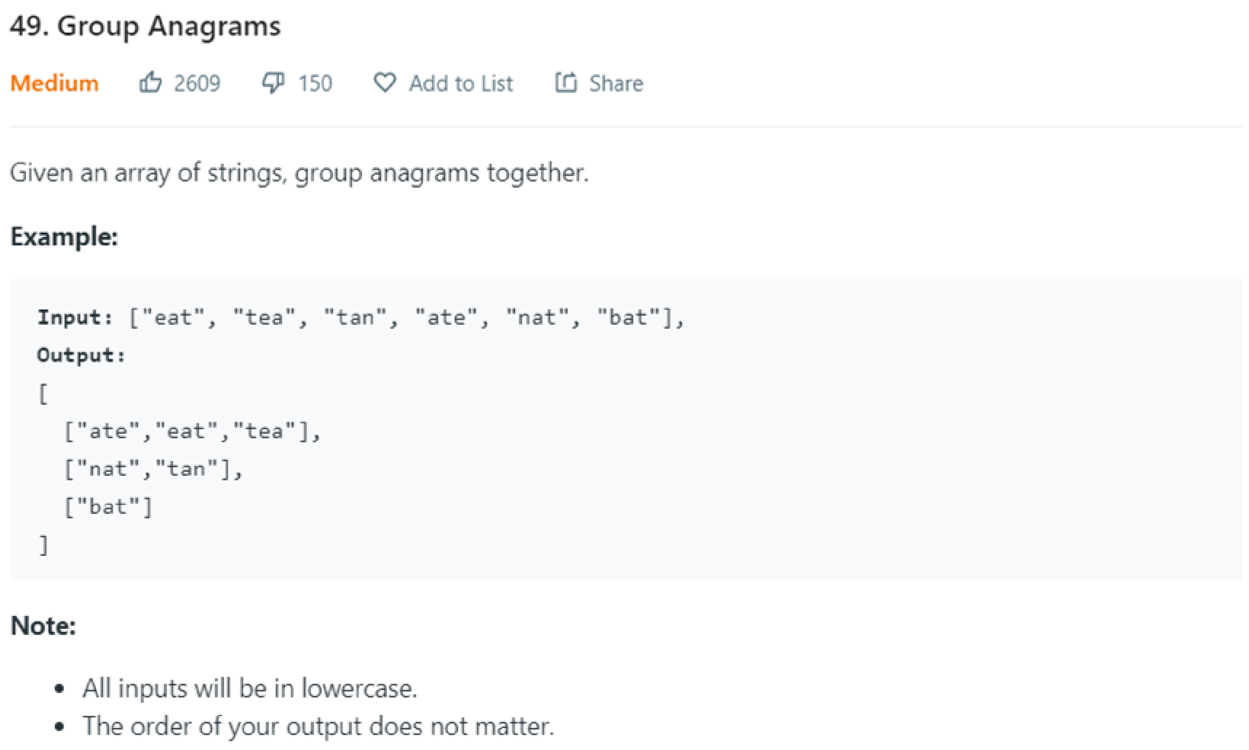
```
canConstruct("a", "b") -> false
canConstruct("aa", "ab") -> false
canConstruct("aa", "aab") -> true
```



Question 2 - Ransom Note

- Idea:
    - Magazine letter count must be >= ransom note letter count
    - Create a frequency map (map letter -> count)
- What’s the time complexity?

Group Anagrams



49. Group Anagrams

Medium 2609 150 Add to List Share

Given an array of strings, group anagrams together.

Example:

```
Input: ["eat", "tea", "tan", "ate", "nat", "bat"],
Output:
[
  ["ate","eat","tea"],
  ["nat","tan"],
  ["bat"]
]
```

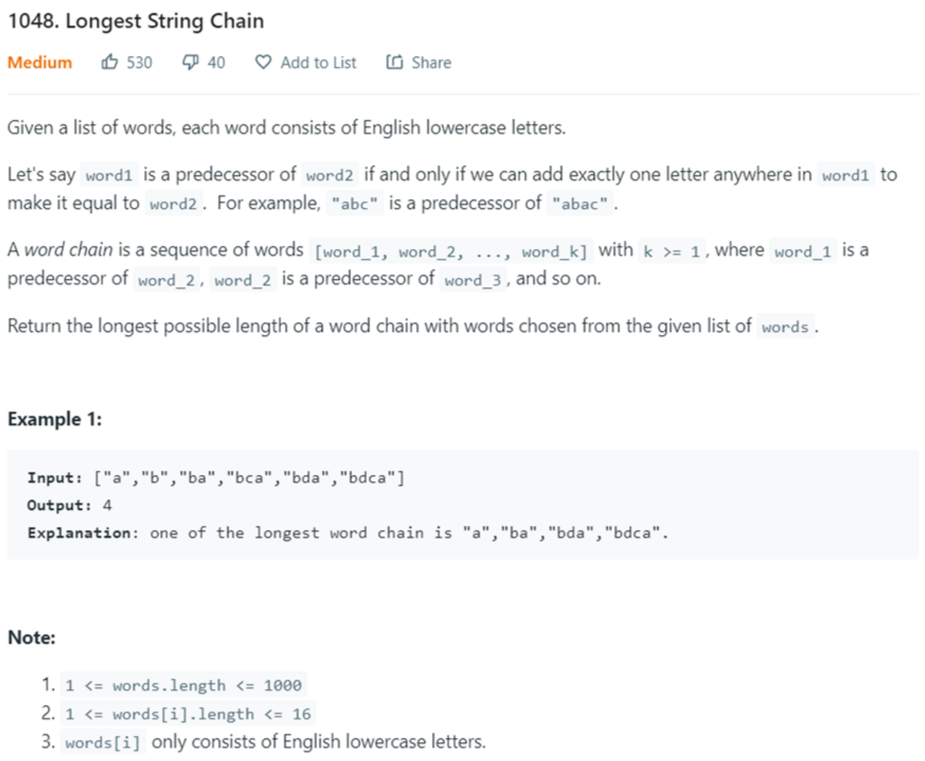
Note:

- All inputs will be in lowercase.
- The order of your output does not matter.

Challenge - Group Anagrams

- What do anagrams have in common?
    - Same number of individual letters (eg: “eat”, “ate”, “eta” each have 1 of each letter)
        - Can we sort the words?
        - Use a hashmap approach?

Extra Challenge (2 Sigma Interview Question)



1048. Longest String Chain

Medium 530 40 Add to List Share

Given a list of words, each word consists of English lowercase letters.

Let's say `word1` is a predecessor of `word2` if and only if we can add exactly one letter anywhere in `word1` to make it equal to `word2`. For example, `"abc"` is a predecessor of `"abac"`.

A *word chain* is a sequence of words `[word_1, word_2, ..., word_k]` with `k >= 1`, where `word_1` is a predecessor of `word_2`, `word_2` is a predecessor of `word_3`, and so on.

Return the longest possible length of a word chain with words chosen from the given list of `words`.

Example 1:

```
Input: ["a","b","ba","bca","bda","bdca"]
Output: 4
Explanation: one of the longest word chain is "a","ba","bda","bdca".
```

Note:

1. `1 <= words.length <= 1000`
2. `1 <= words[i].length <= 16`
3. `words[i]` only consists of English lowercase letters.

https://leetcode.com/problems/longest-string-chain/