

Anagram

An anagram is word switch or word play. It is the result of rearranging the letters of a word or phrase to produce a new word or phrase, while using all the original letters only once.

In interviews, usually we are only bothered with words without spaces in them.

Let's break down the three common approaches in very simple terms:

1. Sorting Both Strings

- **How it works:**

Rearrange (sort) the letters of both strings alphabetically. If the sorted versions are exactly the same, then the strings are anagrams.

- **Time Cost:**

Sorting takes about $O(n \log n)$ time, where n is the number of characters.

- **Space Cost:**

It typically uses extra space proportional to the depth of recursion (about $O(\log n)$).

- **Simple Analogy:**

Imagine you have two mixed-up decks of cards. If you sort them by suit and number, and they end up in the same order, then both decks originally had the same cards.

2. Prime Multiplication Method

- **How it works:**

Assign each letter a unique prime number (like a secret code). For each string, multiply the primes corresponding to its letters. Thanks to the unique nature of prime factors, two strings will have the same product if and only if they have the exact same letters.

- **Time Cost:**

You only go through the string once, so it takes $O(n)$ time.

- **Space Cost:**

It uses constant extra space, $O(1)$, since the mapping of letters to primes is fixed.

- **Simple Analogy:**

Think of it like each letter is a unique ingredient with a special "number." Mix them all together (multiply), and if two recipes (strings) yield the same "flavor number," they contain exactly the same ingredients.

3. Frequency Counting

- **How it works:**

Count how many times each letter appears in each string using a hash (dictionary). Then, compare these counts. If they match for every letter, the strings are anagrams.

- **Time Cost:**

This also takes $O(n)$ time, as you just make one pass through each string.

- **Space Cost:**

It uses constant extra space, $O(1)$, because the number of letters (for example, 26 for the English alphabet) is fixed.

- **Simple Analogy:**

Imagine you have two baskets of fruit. If you count the number of apples, oranges, etc., in both baskets and the counts match exactly, then both baskets have the same mix of fruits.

Each method has its tradeoffs between speed and space:

- **Sorting** is straightforward but slightly slower due to the sorting process.
- **Prime multiplication** is fast and space-efficient, but it can run into issues with very long strings because the multiplication might produce huge numbers.
- **Frequency counting** is both fast and efficient, and it's often the simplest and most reliable method.

Revision #4

Created 28 December 2024 22:46:03 by victor

Updated 17 March 2025 00:18:00 by victor