

Ancient Puzzle - Editorial

Difficulty:

Medium

Prerequisites:

Strings

Hashing - ([Tutorial](#))

Problem in Brief:

You are given a string, S, and a list of words, L, that are all of the same length. Find all starting indices of substring(s) in S that is a concatenation of each word in L exactly once and without any intervening characters.

Editorial:

The basic idea is to iterate through the string S and check whether the current index is a starting point of a concatenation of all words in the list L.

Since we know that all of the words in the given list has the same length, starting from index i, we can cut the string into m substrings, where m is the total number of words in list L, and then check whether each substring is a word in L and is not duplicate.

To check a concatenation, we create a hash that contains all words in L. Each time when a substring matches a word, remove the word from the hash.

Ancient Puzzle - Editorial

Instead of using `unordered_map <string, bool>`, use `unordered_map <string, int>` to record the frequencies of each word. And when we hit a word in the hash, reduce its frequency until it gets down to zero (in that case, the key would have been removed from the hash.).

Time Complexity:

If we Compute the hash of all strings it take $O(k * m + N)$ time and the string search is also linear in time, Hence Complexity is

$O(K * M + N)$

Similar Problems:

[First](#)