Grokking the Coding Interview: Patterns for Coding Questions

95% completed

Search Course

Introduction

Pattern: Sliding Window

Pattern: Two Pointers

Pattern: Fast & Slow pointers

Pattern: Merge Intervals

Pattern: Cyclic Sort

Pattern: In-place Reversal of a LinkedList

Pattern: Tree Breadth First Search

Pattern: Tree Depth First Search

Pattern: Two Heaps

Pattern: Subsets

Pattern: Modified Binary Search

Pattern: Bitwise XOR

Pattern: Top 'K' Elements

Pattern: K-way merge

Pattern : 0/1 Knapsack (Dynamic Programming)

Pattern: Topological Sort (Graph)

- Introduction
- Topological Sort (medium)
- Tasks Scheduling (medium)
- Tasks Scheduling Order (medium)
- All Tasks Scheduling Orders (hard)
- Alien Dictionary (hard)

Alien Dictionary (hard)

We'll cover the following

- Problem Statement
- Try it yourself
- Solution
  - Code
  - Time complexity
  - Space complexity

Problem Statement

There is a dictionary containing words from an alien language for which we don't know the ordering of the characters. Write a method to find the correct order of characters in the alien language.

Example 1:

```
Input: Words: ["ba", "bc", "ac", "cab"]
Output: bac
Explanation: Given that the words are sorted lexicographically by the rules of the alien langua
ge, so
from the given words we can conclude the following ordering among its characters:

1. From "ba" and "bc", we can conclude that 'a' comes before 'c'.
2. From "bc" and "ac", we can conclude that 'b' comes before 'a'

From the above two points, we can conclude that the correct character order is: "bac"
```

Example 2:

```
Input: Words: ["cab", "aaa", "aab"]
Output: cab
Explanation: From the given words we can conclude the following ordering among its characters:

1. From "cab" and "aaa", we can conclude that 'c' comes before 'a'.
2. From "aaa" and "aab", we can conclude that 'a' comes before 'b'

From the above two points, we can conclude that the correct character order is: "cab"
```

Example 3:

```
Input: Words: ["ywx", "wz", "xww", "xz", "zyy", "zwz"]
Output: ywxz
Explanation: From the given words we can conclude the following ordering among its characters:

1. From "ywx" and "wz", we can conclude that 'y' comes before 'w'.
2. From "wz" and "xww", we can conclude that 'w' comes before 'x'.
3. From "xww" and "xz", we can conclude that 'w' comes before 'z'
4. From "xz" and "zyy", we can conclude that 'x' comes before 'z'
5. From "zyy" and "zwz", we can conclude that 'y' comes before 'w'

From the above five points, we can conclude that the correct character order is: "ywxz"
```

Try it yourself

Try solving this question here:

Java | Python3 | JS | C++

```
const find_order = function(words) {
  // TODO: Write your code here
  return "";
};


console.log(`Character order: ${find_order(["ba", "bc", "ac", "cab"])}`)
console.log(`Character order: ${find_order(["cab", "aaa", "aab"])}`)
console.log(`Character order: ${find_order(["ywx", "wz", "xww", "xz", "zyy", "zwz"])}`)

```

RUN SAVE RESET

Solution

Since the given words are sorted lexicographically by the rules of the alien language, we can always compare two adjacent words to determine the ordering of the characters. Take Example-1 above: [“ba”, “bc”, “ac”, “cab”]

1. Take the first two words “ba” and “bc”. Starting from the beginning of the words, find the first character

Problem Challenge 1
Solution Review: Problem Challenge 1
Problem Challenge 2
Solution Review: Problem Challenge 2

Miscellaneous

Kth Smallest Number (hard)

Conclusions

Where to Go from Here

Mark Course as Completed

that is different in both words: it would be ‘a’ from “ba” and ‘c’ from “bc”. Because of the sorted order of words (i.e. the dictionary!), we can conclude that ‘a’ comes before ‘c’ in the alien language.

2. Similarly, from “bc” and “ac”, we can conclude that ‘b’ comes before ‘a’.

These two points tell us that we are actually asked to find the topological ordering of the characters, and that the ordering rules should be inferred from adjacent words from the alien dictionary.

This makes the current problem similar to Tasks Scheduling Order, the only difference being that we need to build the graph of the characters by comparing adjacent words first, and then perform the topological sort for the graph to determine the order of the characters.

Code

Here is what our algorithm will look like (only the highlighted lines have changed):

Java | Python3 | C++ | JS

```
const Deque = require('./collections/deque'); //http://www.collectionsjs.com


function find_order(words) {
  if (words.length === 0) {
    return '';
  }

  // a. Initialize the graph
  const inDegree = {}; // count of incoming edges
  const graph = {}; // adjacency list graph

  words.forEach((word) => {
    for (let i = 0; i < word.length; i++) {
      inDegree[word[i]] = 0;
      graph[word[i]] = [];
    }
  });

  // b. Build the graph
  for (i = 0; i < words.length - 1; i++) { // find ordering of characters from adjacent words
    let w1 = words[i],
      w2 = words[i + 1];
    for (j = 0; j < Math.min(w1.length, w2.length); j++) {
      let parent = w1[j],
        child = w2[j];
      if (parent !== child) { // if the two characters are different
        // put the child into it's parent's list
```

RUN | SAVE | RESET

Close

Output 3.905s

```
Character order: bac
Character order: cab
Character order: ywxz
```

Time complexity

In step ‘d’, each task can become a source only once and each edge (a rule) will be accessed and removed once. Therefore, the time complexity of the above algorithm will be $O(V + E)$, where ‘V’ is the total number of different characters and ‘E’ is the total number of the rules in the alien language. Since, at most, each pair of words can give us one rule, therefore, we can conclude that the upper bound for the rules is $O(N)$ where ‘N’ is the number of words in the input. So, we can say that the time complexity of our algorithm is $O(V + N)$.

Space complexity

The space complexity will be $O(V + N)$, since we are storing all of the rules for each character in an adjacency list.