

Algorithms

DFS Homeowrk 3

Mostafa S. Ibrahim

Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher

PhD from Simon Fraser University - Canada

Bachelor / Msc from Cairo University - Egypt

Ex-(Software Engineer / ICPC World Finalist)



Notes

- These problems might be solved in different ways
- Please, only provide a graph-based treatment
 - This is actually a big hint. Without this hint, you may take a long time to notice a relationship with graphs!

Problem #1: [LeetCode 1743](#) - Restore the Array From Adjacent Pairs

There is an integer array `nums` that consists of `n` **unique** elements, but you have forgotten it. However, you do remember every pair of adjacent elements in `nums`.

You are given a 2D integer array `adjacentPairs` of size `n - 1` where each `adjacentPairs[i] = [ui, vi]` indicates that the elements `ui` and `vi` are adjacent in `nums`.

It is guaranteed that every adjacent pair of elements `nums[i]` and `nums[i+1]` will exist in `adjacentPairs`, either as `[nums[i], nums[i+1]]` or `[nums[i+1], nums[i]]`. The pairs can appear **in any order**.

Return the original array `nums`. If there are multiple solutions, return **any of them**.

- `vector<int> restoreArray(vector<vector<int>> &pairs)`
 - $-10^5 \leq \text{nums}[i], u_i, v_i \leq 10^5$

Example 1:

Input: adjacentPairs = [[2,1],[3,4],[3,2]]

Output: [1,2,3,4]

Explanation: This array has all its adjacent pairs in adjacentPairs.
Notice that adjacentPairs[i] may not be in left-to-right order.

Example 2:

Input: adjacentPairs = [[4,-2],[1,4],[-3,1]]

Output: [-2,4,1,-3]

Explanation: There can be negative numbers.
Another solution is [-3,1,4,-2], which would also be accepted.

Example 3:

Input: adjacentPairs = [[100000,-100000]]

Output: [100000,-100000]

Problem #2: [LeetCode 1202](#) - Smallest String With Swaps

You are given a string `s`, and an array of pairs of indices in the string `pairs` where `pairs[i] = [a, b]` indicates 2 indices(0-indexed) of the string.

You can swap the characters at any pair of indices in the given `pairs` **any number of times**.

Return the **lexicographically smallest** string that `s` can be changed to after using the swaps.

- `string smallestStringWithSwaps(string str, vector<vector<int>> &pairs)`
 - $1 \leq s.length \leq 10^5$
 - $0 \leq pairs.length \leq 10^5$
 - $0 \leq pairs[i][0], pairs[i][1] < s.length$
 - `s` only contains lower case English letters.

Example 1:

Input: s = "dcab", pairs = [[0,3],[1,2]]

Output: "bacd"

Explanation:

Swap s[0] and s[3], s = "bcad"

Swap s[1] and s[2], s = "bacd"

Example 2:

Input: s = "dcab", pairs = [[0,3],[1,2],[0,2]]

Output: "abcd"

Explanation:

Swap s[0] and s[3], s = "bcad"

Swap s[0] and s[2], s = "acbd"

Swap s[1] and s[2], s = "abcd"

Example 3:

Input: s = "cba", pairs = [[0,1],[1,2]]

Output: "abc"

Explanation:

Swap s[0] and s[1], s = "bca"

Swap s[1] and s[2], s = "bac"

Swap s[0] and s[1], s = "abc"

Problem #3: Leetcode [128](#) - Longest Consecutive Sequence

- Given an **unsorted** array of integers, return the length of the longest consecutive elements sequence.
- Input \Rightarrow Output
 - [100, 4, 200, 1, 3, 2] \Rightarrow 4
 - Sequence [4, 1, 3, 2] when sorted [1, 2, 3, 4] \Rightarrow consecutive elements
 - [0, 3, 7, 2, 5, 8, 4, 6, 1] \Rightarrow 9
 - The whole array when sorted is consecutive elements [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- `int longestConsecutive(vector<int>& nums)`
 - $0 \leq \text{nums.length} \leq 10^5$
 - $-10^9 \leq \text{nums}[i] \leq 10^9$

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”