

Grokking the Coding Interview: Patterns for Coding Questions

81% completed



Pattern: Tree Depth First Search



Pattern: Two Heaps



Pattern: Subsets



Pattern: Modified Binary Search



Pattern: Bitwise XOR



Pattern: Top 'K' Elements



- Introduction
- Top 'K' Numbers (easy)
- Kth Smallest Number (easy)
- 'K' Closest Points to the Origin (easy)
- Connect Ropes (easy)
- Top 'K' Frequent Numbers (medium)
- Frequency Sort (medium)
- Kth Largest Number in a Stream (medium)
- 'K' Closest Numbers (medium)
- Maximum Distinct Elements (medium)
- Sum of Elements (medium)
- Rearrange String (hard)
- Problem Challenge 1
- Solution Review: Problem Challenge 1
- Problem Challenge 2
- Solution Review: Problem Challenge 2
- Problem Challenge 3
- Solution Review: Problem Challenge 3

Pattern: K-way merge



- Introduction
- Merge K Sorted Lists (medium)
- Kth Smallest Number in M Sorted Lists (Medium)
- Kth Smallest Number in a Sorted Matrix (Hard)
- Smallest Number Range (Hard)
- Problem Challenge 1
- Solution Review: Problem Challenge 1

Pattern : 0/1 Knapsack (Dynamic Programming)



- Introduction
- 0/1 Knapsack (medium)
- Equal Subset Sum Partition (medium)

Solution Review: Problem Challenge 1

We'll cover the following



- Rearrange String K Distance Apart (hard)
- Solution
- Code
 - Time complexity
 - Space complexity

Rearrange String K Distance Apart (hard)

Given a string and a number 'K', find if the string can be rearranged such that the same characters are at least 'K' distance apart from each other.

Example 1:

```
Input: "mmp", K=2
Output: "mpmp" or "pmpm"
Explanation: All same characters are 2 distance apart.
```

Example 2:

```
Input: "Programming", K=3
Output: "rgmPrgmiano" or "gmrngmrPoa" or "gmrPagimnor" and a few more
Explanation: All same characters are 3 distance apart.
```

Example 3:

```
Input: "aab", K=2
Output: "aba"
Explanation: All same characters are 2 distance apart.
```

Example 4:

```
Input: "aappa", K=3
Output: ""
Explanation: We cannot find an arrangement of the string where any two 'a' are 3 distance apart.
```

Solution

This problem follows the [Top 'K' Numbers](#) pattern and is quite similar to [Rearrange String](#). The only difference is that in the 'Rearrange String' the same characters need not be adjacent i.e., they should be at least '2' distance apart (in other words, there should be at least one character between two same characters), while in the current problem, the same characters should be 'K' distance apart.

Following a similar approach, since we were inserting a character back in the heap in the next iteration, in this problem, we will re-insert the character after 'K' iterations. We can keep track of previous characters in a queue to insert them back in the heap after 'K' iterations.

Code

Here is what our algorithm will look like:

```
Java Python3 C++ JS
1 const Heap = require('./collections/heap'); //http://www.collectionsjs.com
2 const Deque = require('./collections/deque'); //http://www.collectionsjs.com
3
4
5 function reorganize_string(str, k) {
6   if (k <= 1) {
7     return str;
8   }
9
10  charFrequencyMap = {};
11  for (i = 0; i < str.length; i++) {
12    const chr = str[i];
13    if (!(chr in charFrequencyMap)) {
14      charFrequencyMap[chr] = 1;
15    } else {
16      charFrequencyMap[chr]++;
17    }
18  }
19
20
21  const maxHeap = new Heap([], null, ((a, b) => a[0] - b[0]));
22  // add all characters to the max heap
23  Object.keys(charFrequencyMap).forEach((char) => {
24    maxHeap.push([charFrequencyMap[char], char]);
25  });
26
```

Explore

Tracks

My Courses

Edpresso

Refer a Friend

Create

Subset Sum (medium)

Minimum Subset Sum Difference (hard)

Problem Challenge 1

Solution Review: Problem Challenge 1

Problem Challenge 2

Solution Review: Problem Challenge 2

Pattern: Topological Sort (Graph)

Introduction

Topological Sort (medium)

Tasks Scheduling (medium)

Tasks Scheduling Order (medium)

All Tasks Scheduling Orders (hard)

Alien Dictionary (hard)

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

27const queue = new Deque();

28const resultString = [];

RUN

SAVE

RESET

Close

Output4.967s

Reorganized string: mpmp

Reorganized string: rgmorgmaPin

Reorganized string: aba

Reorganized string:

Time complexity

The time complexity of the above algorithm is $O(N * \log N)$ where 'N' is the number of characters in the input string.

Space complexity

The space complexity will be $O(N)$, as in the worst case, we need to store all the 'N' characters in the HashMap.

Interviewing soon? We've partnered with Hired so that companies apply to you instead of you applying to them. [See how](#)

Back

Next

Problem Challenge 1

Problem Challenge 2

Mark as Completed

Report an Issue

Ask a Question