

☆ Premium

ref=nb_npl)





5893. Smallest K-Length Subsequence With Occurrences of a Letter

My Submissions (/contest/weekly-contest-261/problems/smallest-k-length-subsequence-with-occurrences-of-a-letter/submissions/)

Back to Contest (/contest/weekly-contest-261/)

You are given a string s, an integer k, a letter letter, and an integer repetition.

Return the lexicographically smallest subsequence of s of length k that has the letter letter appear at least repetition times. The test cases are generated so that the letter appears in s at least repetition times.

A subsequence is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters.

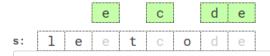
A string a is lexicographically smaller than a string b if in the first position where a and b differ, string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

Example 1:

```
Input: s = "leet", k = 3, letter = "e", repetition = 1
Output: "eet"
Explanation: There are four subsequences of length 3 that have the letter 'e' appear at least 1 time:
- "lee" (from "<u>lee</u>t")
- "let" (from "leet")
- "let" (from "<u>l</u>e<u>et</u>")
- "eet" (from "leet")
The lexicographically smallest subsequence among them is "eet".
```

Example 2:



Input: s = "leetcode", k = 4, letter = "e", repetition = 2 Output: "ecde" Explanation: "ecde" is the lexicographically smallest subsequence of length 4 that has the letter "e" appear at least 2

Example 3:

```
Input: s = "bb", k = 2, letter = "b", repetition = 2
Output: "bb"
Explanation: "bb" is the only subsequence of length 2 that has the letter "b" appear at least 2 times.
```

Constraints:

- 1 <= repetition <= $k <= s.length <= 5 * 10^4$
- s consists of lowercase English letters.
- letter is a lowercase English letter, and appears in s at least repetition times.

```
JavaScript
                                                                                                                        \mathcal{C}
1 \cdot | const smallestSubsequence = (s, k, letter, repetition) => {
        let n = s.length, st = '', letterCnt = 0, len = 0;
2
        let visit = Array(n + 1).fill(0);
3
4 ▼
        for (const c of s) {
5
             if (c == letter) letterCnt++;
6
        for (let i = 0; i < n; i++) {
```

```
8 ▼
              while (1) {}
9
                   if (st.length == 0) break;
10
                   if (st.length + (n - i) \le k) break;
                  if (st[st.length - 1] <= s[i]) break;
if (st[st.length - 1] == letter && len - 1 + letterCnt < repetition) break;
if (st[st.length - 1] == letter) len--;</pre>
11
12
13
                   st = st.slice(0, st.length - 1);
14
15
16
              st += s[i];
17 ▼
              if (s[i] == letter) {
18
                   len++;
19
                   letterCnt--;
              }
20
21
22
         let pos = len - repetition;
23
         len = st.length - k;
24 ▼
         for (let i = st.length - 1; \sim i; i--) {
25 ▼
              if (len > 0 && pos > 0 && st[i] == letter) {
26
                   visit[i] = 1;
27
                   len--;
                   pos--;
28
29
              if (len > 0 && st[i] != letter) {
30 ▼
31
                   visit[i] = 1;
32
                   len--;
33
34
35
         let res = '';
         for (let i = 0; i < st.length; i++) {
36 ▼
37
              if (!visit[i]) res += st[i];
38
39
         return res;
    };
40
```

□ Custom Testcase

Use Example Testcases

Submission Result: Accepted (/submissions/detail/564941122/) ?

More Details ➤ (/submissions/detail/564941122/)

Run

Share your acceptance!

Copyright © 2021 LeetCode

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)