

6099. Longest Binary Subsequence Less Than or Equal to K

My Submissions (/contest/weekly-contest-298/problems/longest-binary-subsequence-less-than-or-equal-to-k/submissions/)

Back to Contest (/contest/weekly-contest-298/)

You are given a binary string `s` and a positive integer `k`.

Return the length of the **longest** subsequence of `s` that makes up a **binary** number less than or equal to `k`.

Note:

- The subsequence can contain **leading zeroes**.
- The empty string is considered to be equal to `0`.
- 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.

| | |
|--------------------|--------|
| User Accepted: | 0 |
| User Tried: | 0 |
| Total Accepted: | 0 |
| Total Submissions: | 0 |
| Difficulty: | Medium |

Example 1:

Input: `s = "1001010"`, `k = 5`
Output: `5`
Explanation: The longest subsequence of `s` that makes up a binary number less than or equal to 5 is "00010", as this number is equal to 5 in decimal. Note that "00100" and "00101" are also possible, which are equal to 4 and 5 in decimal, respectively. The length of this subsequence is 5, so 5 is returned.

Example 2:

Input: `s = "00101001"`, `k = 1`
Output: `6`
Explanation: "000001" is the longest subsequence of `s` that makes up a binary number less than or equal to 1, as this number is equal to 1 in decimal. The length of this subsequence is 6, so 6 is returned.


Constraints:

- `1 <= s.length <= 1000`
- `s[i]` is either `'0'` or `'1'`.
- `1 <= k <= 109`

JavaScript

📄 ↺ ⚙

```
1 const longestSubsequence = (s, k) => {
2   let t = k.toString(2), m = t.length, n = s.length, pre = Array(n + 1).fill(0);
3   for (let i = 0; i < n; i++) pre[i + 1] = pre[i] + (s[i] ^ 1);
4   let res = pre[n];
5   for (let i = 0; i < n; i++) {
6     if (s[i] == '1') {
7       res = Math.max(res, pre[i] + Math.min(m - 1, n - i));
8       let p = 0;
9       for (let j = i; j < n && p < m; j++) {
10        if (t[p] == '1' && s[j] == '0') {
11          p = Math.min(m, p + (n - j));
12          break;
13        }
14        if (t[p] == '0' && s[j] == '1') {
15        } else {
16          p++;
17        }
18      }
19      res = Math.max(res, pre[i] + p);
20    }
21  }
22  return res;
23 }
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)Submission Result: **Accepted** (</submissions/detail/725710523/>) [More Details >](/submissions/detail/725710523/)

Share your acceptance!

[◀ 2](#)

Copyright © 2022 LeetCode

[Help Center \(/support\)](/support/) | [Jobs \(/jobs\)](/jobs/) | [Bug Bounty \(/bugbounty\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student\)](/student/) | [Terms \(/terms\)](/terms/) | [Privacy Policy \(/privacy\)](/privacy/)[United States \(/region\)](/region/)