Previous Next

758. Bold Words In String

Jan. 6, 2018 | 7.2K views

Average Rating: 3.38 (8 votes)

Given a set of keywords `words` and a string `S`, make all appearances of all keywords in `S` bold. Any letters between `<b>` and `</b>` tags become bold.

The returned string should use the least number of tags possible, and of course the tags should form a valid combination.

For example, given that `words = ["ab", "bc"]` and `S = "aabcd"`, we should return `"a<b>abc</b>d"`. Note that returning `"a<b>a<b>b</b>c</b>d"` would use more tags, so it is incorrect.

Note:

1. `words` has length in range `[0, 50]`.
2. `words[i]` has length in range `[1, 10]`.
3. `S` has length in range `[0, 500]`.
4. All characters in `words[i]` and `S` are lowercase letters.

Approach #1: Brute Force [Accepted]

Intuition

Let's try to learn which letters end up bold, since the resulting answer will just be the canonical one - we put bold tags around each group of bold letters.

To do this, we'll check for all occurrences of each word and mark the corresponding letters bold.

Algorithm

Let's work on first setting `mask[i] = true` if and only if the `i`-th letter is bold. For each starting position `i` in `S`, for each `word`, if `S[i]` starts with `word`, we'll set the appropriate letters bold.

Now armed with the correct `mask`, let's try to output the answer. A letter in position `i` is the first bold letter of the group if `mask[i] && (i == 0 || !mask[i-1])`, and is the last bold letter if `mask[i] && (i == N-1 || !mask[i+1])`. Alternatively, we could use `itertools.groupby` in Python.

Once we know which letters are the first and last bold letters of a group, we know where to put the `"<b>"` and `"</b>"` tags.

Java | Python | Copy

```
class Solution(object):
    def boldWords(self, words, S):
        N = len(S)
        mask = [False] * N
        for i in xrange(N):
            prefix = S[i:]
            for word in words:
                if prefix.startswith(word):
                    for j in xrange(i, min(i+len(word), N)):
                        mask[j] = True

        ans = []
        for incl, grp in itertools.groupby(zip(S, mask), lambda z: z[1]):
            if incl: ans.append("<b>")
            ans.append("".join(z[0] for z in grp))
            if incl: ans.append("</b>")
        return "".join(ans)
```

Complexity Analysis

- Time Complexity: $O(N \sum w_i)$, where N is the length of `S` and w_i is the sum of `W`.
- Space Complexity: $O(N)$.

Analysis written by: @awice.

Rate this article:

Previous Next

Comments: 13 Sort By

Type comment here... (Markdown is supported)

Preview Post

ajay13 ★ 120 November 30, 2018 8:38 PM
I think this problem should be tagged medium. Definitely not an easy one.
23 Share Reply
SHOW 2 REPLIES

vkvk123vkvk321 ★ 13 July 31, 2018 1:15 AM
The good ol' rack-your-brain-until-you-realize-brute-force-is-the-only-option question. Run away if your interviewer asks this.
13 Share Reply

GCC97 ★ 73 February 17, 2019 7:49 AM
How is this tagged as an easy problem? I wish leetcode had emojis!
7 Share Reply

vegito2002 ★ 1178 January 18, 2018 3:43 AM
for the java versioN: Get rid of `match` function, you never called it.
6 Share Reply

AmekiKyou ★ 77 June 4, 2018 2:10 PM
It is right to write a brute-force.
You will never have the time to write an Aho-Corasick Automation in interviews LOL.
4 Share Reply

sevenhe716 ★ 70 March 2, 2019 3:24 PM
optimized python solution using trie tree and merge intervals:

1. **trie tree** is used to speed up string match (faster than **find** or **startwith** in large query request).
2. Using **merge intervals** instead of **mask** to reduce Time and Space Complexity, both from O(n) to O(m), m represets interval numbers after merged.

Read More
3 Share Reply

salamanderrex ★ 32 September 17, 2018 3:50 AM
for python

```
for j in xrange(i, min(i+len(word), N)):
    mask[j] = True
```

Read More
2 Share Reply
SHOW 1 REPLY

oddguy ★ 2 December 27, 2018 9:48 PM
In fact you don't have to mask EVERY letter to true! A "interval border addition" method can works for this problem. Looks like it reduces the Time complexcity to O(N) (just ignoring the complexity of using indexOf()).
1 Share Reply
SHOW 1 REPLY

leetcodefan ★ 1503 May 3, 2019 11:38 AM
Seriously, Leetcode #616 `Add Bold Tag In String` is marked as medium while this one is "easy"........
0 Share Reply

calvinchankf ★ 2515 April 20, 2019 12:05 PM
i spent so much time on finding out a better approach and u now tell me that u actually accept the brute approach 😑
0 Share Reply

‹ 1 2 ›