

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(ns + bs) time | O(ns) space
6 func MultiStringSearch(bigString string, smallStrings []string) []bool {
7     trie := Trie{children: map[byte]Trie{}}
8     for _, str := range smallStrings {
9         trie.Add(str)
10    }
11    containedStrings := map[string]bool{}
12    for i := range bigString {
13        findSmallStringsIn(bigString, i, trie, containedStrings)
14    }
15    output := make([]bool, len(smallStrings))
16    for i, str := range smallStrings {
17        output[i] = containedStrings[str]
18    }
19    return output
20 }
21
22 func findSmallStringsIn(str string, startIdx int, trie Trie, contained
23     current := trie
24     for i := startIdx; i < len(str); i++ {
25         currentChar := str[i]
26         if _, found := current.children[currentChar]; !found {
27             break
28         }
29         current = current.children[currentChar]
30         if end, found := current.children['*']; found {
31             containedStrings[end.word] = true
32         }
33     }
```

Solution 1

Solution 2

Solution 3

```
1 package main
2
3 func MultiStringSearch(bigString string, smallStrings []string) []bool {
4     // Write your code here.
5     return nil
6 }
7
```

Our Tests

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

```
14 Run in Jupyter Notebook Notebook 1
15 expected = ["Red", "Blue", "Blue", "Blue", "Blue", "Blue", "Blue"]
16 output = Reorderingsolver("Blue is a big string", ["string", "Red"])
17 Reorderingsolver(expected, output)
18 }
19
20 Run in Jupyter Notebook Notebook 1
21 expected = ["Red", "Blue", "Blue", "Blue", "Blue", "Blue", "Blue"]
22 output = Reorderingsolver("Blue goes to the shopping center early")
23 Reorderingsolver(expected, output)
24 }
25
26 Run in Jupyter Notebook Notebook 1
27 expected = ["Red", "Blue", "Blue", "Blue", "Blue", "Blue", "Blue"]
```

Run or submit code when you're ready.