

Backspace String Compare (/problems/backspace-string-compare/)

Submission Detail

104 / 104 test cases passed.

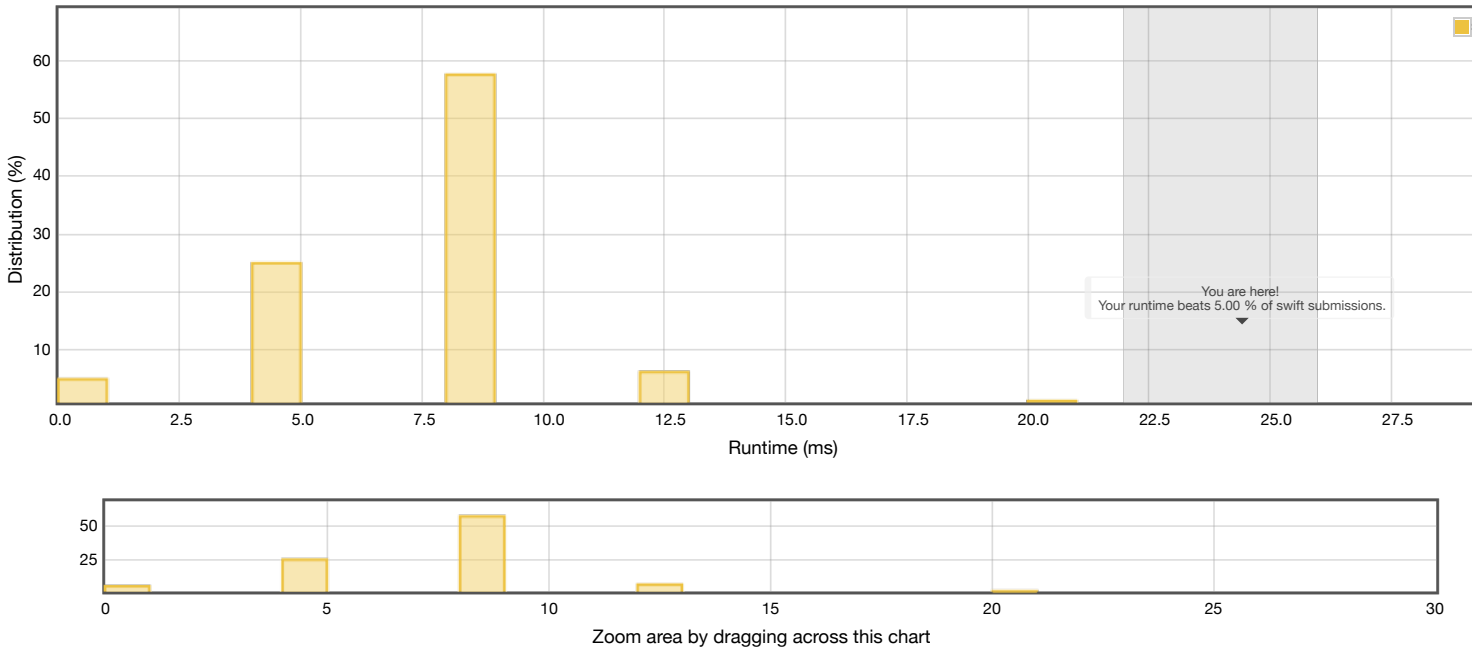
Runtime: 24 ms

Memory Usage: 21.9 MB

Status: Accepted

Submitted: 0 minutes ago

Accepted Solutions Runtime Distribution



Accepted Solutions Memory Distribution

Sorry. We do not have enough accepted submissions to show distribution chart.

Invite friends to challenge **Backspace String Compare**

Submitted Code: 0 minutes ago

Language: swift

Edit Co

```
1 func myPrint(_ arg: Any) {
2     //print(arg)
3 }
4
5 class Solution {
6     func backspaceCompare(_ S: String, _ T: String) -> Bool {
7
8         assert(S.count > 0)
9         assert(T.count > 0)
10
11         var idxS = S.count-1
12         var idxT = T.count-1
13
14         while true {
15             let charS: String?
16             let charT: String?
17             (charS, idxS) = nextFreeChar(S, index: idxS)
18             myPrint("S: \(String(describing: charS)), \(idxS)")
19             (charT, idxT) = nextFreeChar(T, index: idxT)
20             myPrint("T: \(String(describing: charT)), \(idxT)")
21         }
```

```
22         if charS == nil && charT != nil { return false }
23         if charS != nil && charT == nil { return false }
24         if idxS == -1 && idxT == -1 { return true}
25         if let charS = charS, let charT = charT, charS != charT { return false }
26     }
27 }
28
29 private func nextFreeChar(_ str: String, index: Int) -> (String?, Int) {
30     var charsToExplore = 1
31     var currentIndex = index
32     var currentChar: String? = nil
33
34     while charsToExplore > 0 && currentIndex >= 0 {
35         currentChar = str.string(at: currentIndex)!
36         if currentChar == "#" {
37             charsToExplore += 2
38         }
39         currentIndex -= 1
40         charsToExplore -= 1
41     }
42
43     if charsToExplore > 0 {
44         return (nil, currentIndex)
45     } else {
46         return (currentChar, currentIndex)
47     }
48 }
49 }
50
51 public extension String {
52     func index(at i: Int) -> String.Index? {
53         guard i < self.count && i >= 0 else {
54             return nil
55         }
56         var index = self.startIndex
57         formIndex(&index, offsetBy: i)
58         return index
59     }
60
61     func string(at i: Int) -> String? {
62         guard let index = index(at: i) else {
63             return nil
64         }
65
66         return String(self[index])
67     }
68 }
```

[Back to problem \(/problems/backspace-string-compare/\)](/problems/backspace-string-compare/)