Contiguous Array

Submission Detail

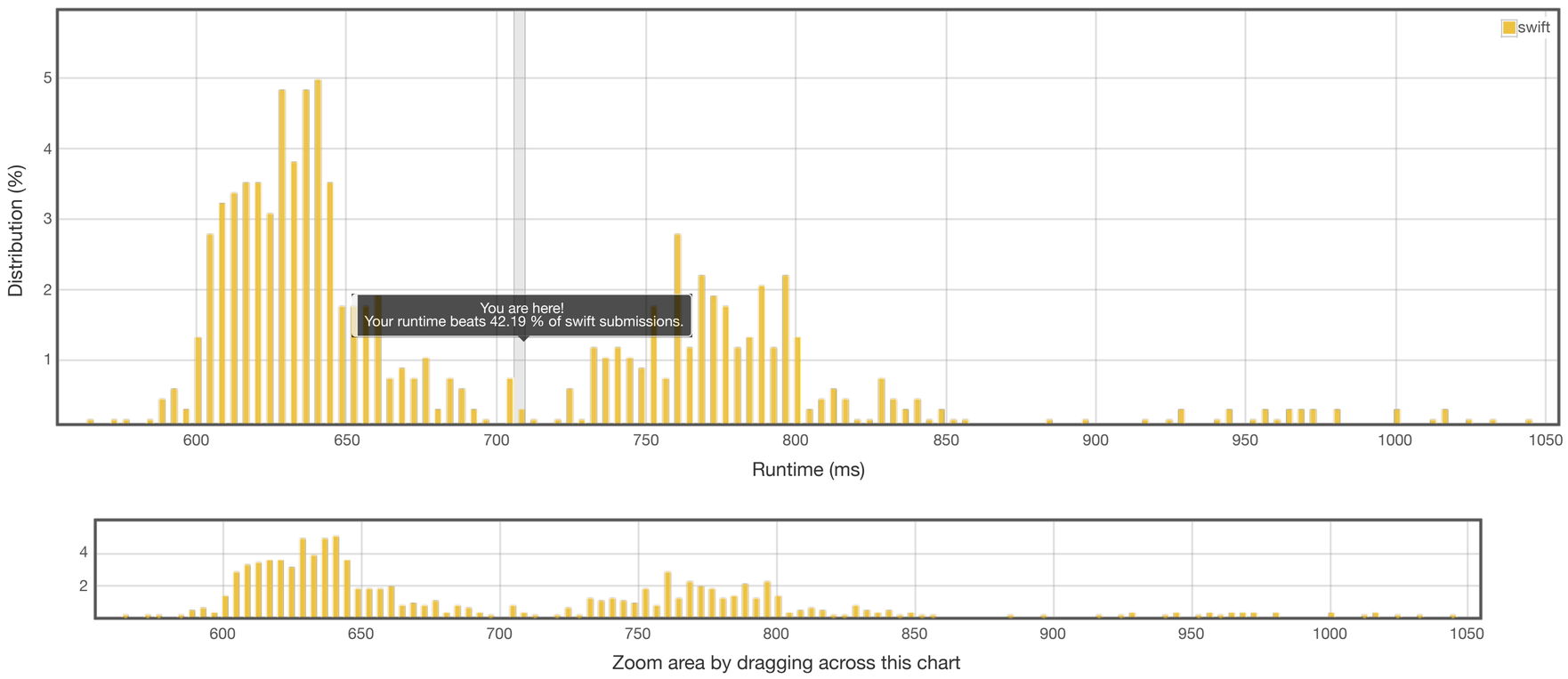
555 / 555 test cases passed.

Runtime: **708 ms**
Memory Usage: **25.9 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 **Contiguous Array**

13

Submitted Code: 0 minutes ago

Language: swift

Edit Code

```swift

class Solution {
    func findMaxLengthWithDict(_ nums: [Int]) -> Int {
        guard nums.count > 0 else { return 0 }

        var sumToIndicesDict: [Int: (Int, Int?)] = [0:(0,nil)]

        func register(sum: Int, newIndex: Int) {
            if let (i, _) = sumToIndicesDict[sum] {
                sumToIndicesDict[sum] = (i, newIndex)
            } else {
                sumToIndicesDict[sum] = (newIndex, nil)
            }
        }

        func getLength(indices: (Int,Int?)) -> Int? {
            if let j = indices.1 {
                return j - indices.0
            } else {
                return nil
            }
        }

        var i = 0
        var sum = 0
        while i < nums.count {
            sum  = nums[i] == 0 ? sum - 1 : sum + 1
            i += 1
            register(sum: sum, newIndex: i)
        }

        // Only qualifying substrings, i.e. they have a second index
        let lengths = sumToIndicesDict.values.compactMap { indices in
            return getLength(indices: indices)
        }

        return lengths.max() ?? 0
    }

    func findMaxLength(_ nums: [Int]) -> Int {
        return findMaxLengthWithDict(nums)
    }
}
```

Back to problem