560. Subarray Sum Equals K

Hint

Medium 19.6K 575

Companies

Given an array of integers `nums` and an integer `k`, return *the total number of subarrays whose sum equals to* `k`.

A subarray is a contiguous **non-empty** sequence of elements within an array.

Example 1:

```
Input: nums = [1,1,1], k = 2
Output: 2
```

Example 2:

```
Input: nums = [1,2,3], k = 3
Output: 2
```

Constraints:

- $1 <= \text{nums.length} <= 2 * 10^4$
- $-1000 <= \text{nums[i]} <= 1000$
- $-10^7 <= k <= 10^7$

Accepted 1M Submissions 2.4M Acceptance Rate 43.3%

Discussion (80)

Similar Questions

Related Topics

Copyright © 2023 LeetCode All rights reserved

1109. Corporate Flight Bookings

Medium 1.5K 155

Companies

There are `n` flights that are labeled from `1` to `n`.

You are given an array of flight bookings `bookings`, where `bookings[i] = [first_i, last_i, seats_i]` represents a booking for flights `first_i` through `last_i` (**inclusive**) with `seats_i` seats reserved for **each flight** in the range.

Return *an array* `answer` *of length* `n`*, where* `answer[i]` *is the total number of seats reserved for flight* `i`.

Example 1:

```
Input: bookings = [[1,2,10],[2,3,20],[2,5,25]], n = 5
Output: [10,55,45,25,25]
Explanation:
Flight labels:        1   2   3   4   5
Booking 1 reserved:  10  10
Booking 2 reserved:      20  20
Booking 3 reserved:      25  25  25  25
Total seats:         10  55  45  25  25
Hence, answer = [10,55,45,25,25]
```

Example 2:

```
Input: bookings = [[1,2,10],[2,2,15]], n = 2
Output: [10,25]
Explanation:
Flight labels:        1   2
Booking 1 reserved:  10  10
Booking 2 reserved:      15
Total seats:         10  25
Hence, answer = [10,25]
```

Constraints:

- $1 <= n <= 2 * 10^4$
- $1 <= bookings.length <= 2 * 10^4$
- `bookings[i].length == 3`

- `1 <= first_i <= last_i <= n`
- `1 <= seats_i <= 10^4`

Accepted 55.1K Submissions 90.5K Acceptance Rate 60.9%

Discussion (7)

Related Topics

Copyright © 2023 LeetCode All rights reserved

239. Sliding Window Maximum

Hint

Hard 16.9K 570

Companies

You are given an array of integers `nums`, there is a sliding window of size `k` which is moving from the very left of the array to the very right. You can only see the `k` numbers in the window. Each time the sliding window moves right by one position.

Return *the max sliding window*.

Example 1:

```
Input: nums = [1,3,-1,-3,5,3,6,7], k = 3
Output: [3,3,5,5,6,7]
Explanation: 
Window position                Max
---------------               -----
[1  3  -1] -3  5  3  6  7       3
 1 [3  -1  -3] 5  3  6  7       3
 1  3 [-1  -3  5] 3  6  7       5
 1  3  -1 [-3  5  3] 6  7       5
 1  3  -1  -3 [5  3  6] 7       6
 1  3  -1  -3  5 [3  6  7]      7
```

Example 2:

```
Input: nums = [1], k = 1
Output: [1]
```

Constraints:

- `1 <= nums.length <=` 10^5
- -10^4 `<= nums[i] <=` 10^4
- `1 <= k <= nums.length`

Accepted 878.6K Submissions 1.9M Acceptance Rate 46.6%

Discussion (101)

1024. Video Stitching

Hint

Medium 1.6K 55

Companies

You are given a series of video clips from a sporting event that lasted `time` seconds. These video clips can be overlapping with each other and have varying lengths.

Each video clip is described by an array `clips` where `clips[i] = [start_i, end_i]` indicates that the ith clip started at `start_i` and ended at `end_i`.

We can cut these clips into segments freely.

- For example, a clip `[0, 7]` can be cut into segments `[0, 1] + [1, 3] + [3, 7]`.

Return *the minimum number of clips needed so that we can cut the clips into segments that cover the entire sporting event* `[0, time]`. If the task is impossible, return `-1`.

Example 1:

```
Input: clips = [[0,2],[4,6],[8,10],[1,9],[1,5],[5,9]], time = 10
Output: 3
Explanation: We take the clips [0,2], [8,10], [1,9]; a total of 3 clips.
Then, we can reconstruct the sporting event as follows:
We cut [1,9] into segments [1,2] + [2,8] + [8,9].
Now we have segments [0,2] + [2,8] + [8,10] which cover the sporting event [0, 10].
```

Example 2:

```
Input: clips = [[0,1],[1,2]], time = 5
Output: -1
Explanation: We cannot cover [0,5] with only [0,1] and [1,2].
```

Example 3:

```
Input: clips = [[0,1],[6,8],[0,2],[5,6],[0,4],[0,3],[6,7],[1,3],[4,7],[1,4],[2,5],[2,6],[3,4],[4,5],[5,7],[6,9]], time = 9
Output: 3
Explanation: We can take clips [0,4], [4,7], and [6,9].
```

Constraints:

- `1 <= clips.length <= 100`

- `0 <= start_i <= end_i <= 100`
- `1 <= time <= 100`

Accepted 59.7K Submissions 118.1K Acceptance Rate 50.6%

Discussion (8)

Related Topics

Copyright © 2023 LeetCode All rights reserved

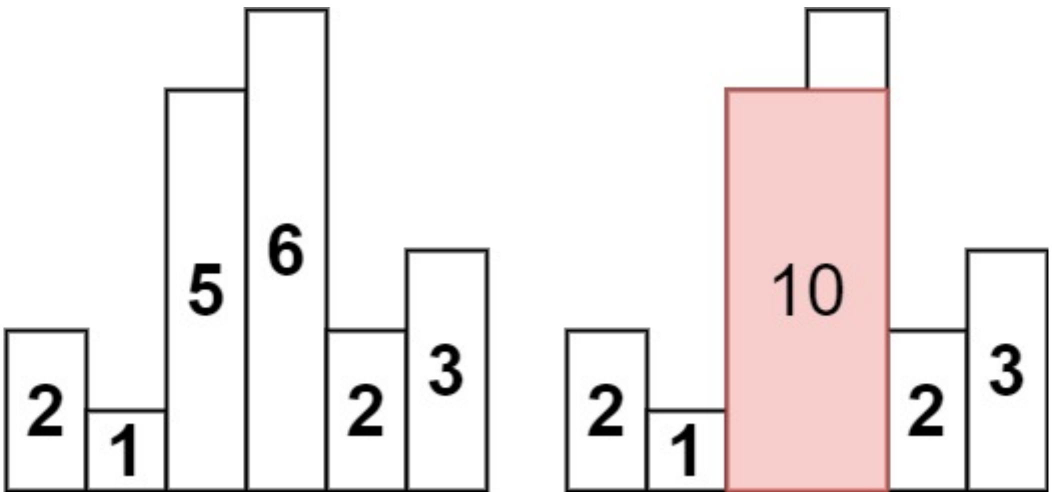
84. Largest Rectangle in Histogram

Hard 15.6K 225

Companies

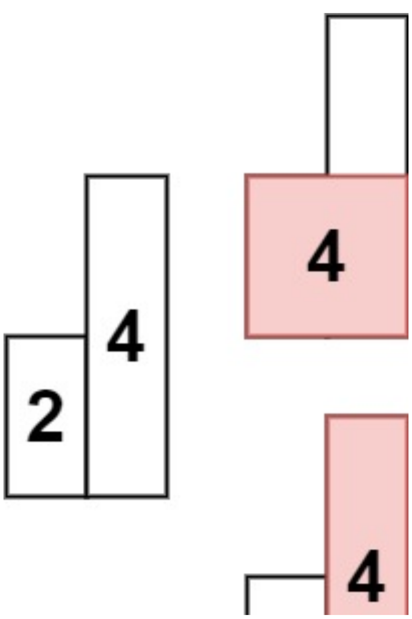
Given an array of integers `heights` representing the histogram's bar height where the width of each bar is `1`, return *the area of the largest rectangle in the histogram*.

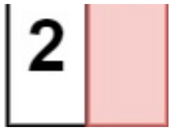
Example 1:




```
Input: heights = [2,1,5,6,2,3]
Output: 10
Explanation: The above is a histogram where width of each bar is 1.
The largest rectangle is shown in the red area, which has an area = 10 units.
```

Example 2:





> **Input:** heights = [2,4]
> **Output:** 4

Constraints:

- `1 <= heights.length <= 10`^{5}
- `0 <= heights[i] <= 10`^{4}

Accepted 730K   Submissions 1.7M   Acceptance Rate 43.2%

Discussion (68)

Similar Questions

Related Topics

Copyright © 2023 LeetCode All rights reserved