Challenge 11: Maximum Sum Sublist

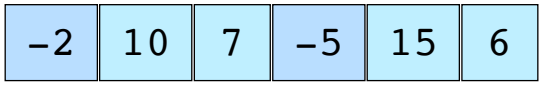
Given an array, find the contiguous sublist with the largest sum.

We'll cover the following

Maximum sublist sum: An overview#

Given an unsorted list A, the maximum sum sub list is the sub list (contiguous elements) from A for which the sum of the elements is maximum. In this challenge, we want to find the sum of the maximum sum sub list. This problem is a tricky one because the list might have negative integers in any position, so we have to cater to those negative integers while choosing the continuous sublist with the largest positive values.

Here is an example:



Brute force method to find maximum subarray sum

1 of 8

In the above slides, we are showing a brute force approach to solve this problem. We are computing the sum of all possible sublists to find the maximum. Your task is to come up with a more efficient technique to solve this problem.

Problem statement#

Given an integer list, return the maximum sublist sum. The l[illegible]y[illegible]ai[illegible] both positive and negative integers and is unsorted.

Partial function definition#

```
def find_max_sum_sublist(lst):
  pass
```

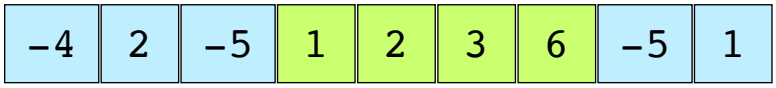
Input#

- a list `lst`

Output#

- a number (maximum subarray sum)

Sample input#



Sample output#

```
largest_sum = 12
```

Coding challenge#

Take a close look and design a step-by-step algorithm before jumping on to the implementation. This problem is designed for your practice, so try to solve it on your own first before referring to the solution ahead.

Good Luck!

```
def find_max_sum_sublist(lst):
  # Write your code here!
  running_sum = 0
  global_sum = lst[0]
  for ele in lst:
    running_sum += ele
    if running_sum < global_sum:
      running_sum = 0
    elif running_sum > global_sum:
      global_sum = running_sum
  return global_sum
```

Let’s go over the solution review of this challenge in the upcoming lesson.

Back

Solution Review: Rearrange Sorted Lis...

Next

Solution Review: Maximum Sum Sublist

Completed

Report an Issue