

# Problem 2599: Make the Prefix Sum Non-negative

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 51.84%

**Paid Only:** Yes

**Tags:** Array, Greedy, Heap (Priority Queue)

## Problem Description

You are given a **0-indexed** integer array `nums`. You can apply the following operation any number of times:

- \* Pick any element from `nums` and put it at the end of `nums`.

The prefix sum array of `nums` is an array `prefix` of the same length as `nums` such that `prefix[i]` is the sum of all the integers `nums[j]` where `j` is in the inclusive range `[0, i]`.

Return the minimum number of operations such that the prefix sum array does not contain negative integers. The test cases are generated such that it is always possible to make the prefix sum array non-negative.

**Example 1:**

**Input:** nums = [2,3,-5,4] **Output:** 0 **Explanation:** we do not need to do any operations. The array is [2,3,-5,4]. The prefix sum array is [2, 5, 0, 4].

**Example 2:**

**Input:** nums = [3,-5,-2,6] **Output:** 1 **Explanation:** we can do one operation on index 1. The array after the operation is [3,-2,6,-5]. The prefix sum array is [3, 1, 7, 2].

**Constraints:**

```
* `1 <= nums.length <= 105` * `-109 <= nums[i] <= 109`
```

## Code Snippets

### C++:

```
class Solution {  
public:  
    int makePrefSumNonNegative(vector<int>& nums) {  
  
    }  
};
```

### Java:

```
class Solution {  
public int makePrefSumNonNegative(int[ ] nums) {  
  
}  
}
```

### Python3:

```
class Solution:  
    def makePrefSumNonNegative(self, nums: List[int]) -> int:
```