



# Smallest Missing Integer Greater Than Sequential Prefix Sum

You are given a **0-indexed** array of integers `nums`.

A prefix `nums[0..i]` is **sequential** if, for all  $1 \leq j \leq i$ ,  $nums[j] = nums[j - 1] + 1$ . In particular, the prefix consisting only of `nums[0]` is **sequential**.

Return the **smallest** integer `x` missing from `nums` such that `x` is greater than or equal to the sum of the **longest** sequential prefix.

## Example 1:

Input: `nums = [1,2,3,2,5]`

Output: 6

Explanation: The longest sequential prefix of `nums` is `[1,2,3]` with a sum of 6. 6 is not in the array, therefore 6 is the smallest missing integer greater than or equal to the sum of the longest sequential prefix.

## Example 2:

Input: `nums = [3,4,5,1,12,14,13]`

Output: 15

Explanation: The longest sequential prefix of `nums` is `[3,4,5]` with a sum of 12. 12, 13, and 14 belong to the array while 15 does not. Therefore 15 is the smallest missing integer greater than or equal to the sum of the longest sequential prefix.

first lets define the variables we are going to need lets first set the count to the first element in the list and i a index of 0.

```
count = nums[0]
```

```
i = 0
```

now lets create our loop logic imagine we want a while loop that loops through all the numbers in the nums array and increments every time.

```
while i < len(nums):
```

during this loop we can create a conditional block to check if the next number is exactly 1 increment above the other one if it is we will increase count if so count is increased by one otherwise we will exit the loop

```
while i < len(nums):  
    i += 1  
    if nums[i]==nums[i-1]+1:  
        count+= nums[i]  
    else:  
        break
```

we will then create another while loop to say if there is no count we will return it as index 0 so for example if we had {1, 3, 5, 6} we would just get one because 1 and 3 have a difference of 2 and not one else increase the count.

```
while True:  
    if count not in nums:  
        return count
```

```
else:  
    count+=1
```

## Complete solution

```
from ast import List  
  
class Solution:  
    def missingInteger(self, nums: List[int]) → int:  
        count = nums[0] # set the count to the first element in the list  
        i = 0 # set the index to 0  
        while i < len(nums):  
            i += 1  
            if nums[i]==nums[i-1]+1:  
                count+= nums[i]  
            else:  
                break  
        while True:  
            if count not in nums:  
                return count  
            else:  
                count+=1
```