Daily Coding Problem #224

Problem

This problem was asked by Amazon.

Given a sorted array, find the smallest positive integer that is not the sum of a subset of the array.

For example, for the input `[1, 2, 3, 10]`, you should return `7`.

Do this in `O(N)` time.

Solution

A naive solution would be to start with `1` and keep incrementing until we find a solution. However, since the problem of determining whether a subset of an array sums to a given number is NP-complete, this approach cannot succeed in linear time.

Looking at the example given above may provide some insight.

Before looking at our list, we know the smallest "impossible sum" is `1`- the lowest positive integer. When we come across `1` as the first element in our array, though, this must change. What should the new impossible sum be? In this case, it's easy to see that it must be `2`.

Let's continue traversing the array. The next element we come across is `2`. Since the impossible sum at the moment is `2`, we know we can make any (non-negative) sum less than that. So by adding our new element to each, we can make two new sums: `0 + 2` and `1 + 2`. This motivates us to increase our impossible sum to `4`.

Now we move on to the third element, `3`. Again, since our current impossible sum is `4`, we are already able to make sums from `0` through `3`. Adding `3` to each of these generates three new sums: `0 + 3`, `1 + 3`, `2 + 3`, and `3 + 3`, and allows us to bump the impossible sum to `7`.

Finally, since the next element is `10`, it will be of no help in creating a subset that sums to `7`, and so we have our answer.

The pattern above is as follows: as we traverse the array, whenever we see a value less than or equal to our current impossible sum, we increment the impossible sum by that value. Once we find a value greater than the impossible sum, we return that sum.

```
def smallest_impossible_sum(nums):
    impossible_sum = 1
    for n in nums:
        if n <= impossible_sum:
            impossible_sum += n
        else:
            break
    return impossible_sum
```