Problem of the Week

Problem Title: *Smallest Non-Representable Sum*

problem asked by Amazon

Problem Description:

Scenario:
You are designing a secure digital wallet system. Each user has a set of coin denominations (represented as a sorted array of positive integers). For internal validation, you must determine the **smallest amount of money** that cannot be formed using **any subset** of the given denominations.

This functionality is crucial for detecting missing denominations and optimizing wallet design. The challenge? You need to solve this **efficiently** – in linear time relative to the size of the input array.

Input Format:

- A single line containing space-separated **sorted** positive integers:
  `a1 a2 a3 ... an`

Output Format:

- Print a single integer: the **smallest positive integer** that cannot be formed as the sum of any subset of the array.

Constraints:

- `1 <= N <= 10^5`
- `1 <= a[i] <= 10^9`
- The input array is sorted in **increasing order**
- All elements are **positive integers**

Example Input:

```
1 2 3 10
```

Example Output:

```
7
```

Explanation:

With the given array `[1, 2, 3, 10]`:

- We can make: 1, 2, 3, 1+2=3, 1+3=4, 2+3=5, 1+2+3=6
- But we **cannot** make **7** — it's the smallest number that's not representable using any subset of the array.

So, the answer is `7`.

Hint:

Use a **greedy approach**:

- Start with the smallest number that cannot be formed yet (initially 1).
- Traverse the array and keep extending the range of constructible sums.
- If the current number in the array is greater than the current smallest non-representable number, then that number is the answer.

Optimal Time Complexity:

O(N) — One linear pass through the array.

Practice Links:

- 🔗 GeeksforGeeks – Smallest Positive Integer that cannot be represented as sum
- 🔗 Leetcode (Similar): Partition Equal Subset Sum

Video Solutions:

- 📹 GFG YouTube – Smallest number that cannot be represented as subset sum