Not A Subset Sum

solved by	Senan
Platform	GeeksForGeeks
difficulty	Medium
tags	Logic Sorting
language	C++
solved on	@05/10/2024
link	https://www.geeksforgeeks.org/problems/smallest-number-subset1220/1
Completion	✓

Intuition

The problem involves finding the smallest positive integer that cannot be represented as the sum of any subset of a given array. The intuition behind this is that if we can form all numbers from 1 to `sum - 1` using a subset of the array, but we encounter an element larger than `sum`, then this element will create a gap in the possible sums. Thus, the smallest integer we can no longer form is `sum`.

Approach

1. Start by initializing `sum = 1`. This represents the smallest integer that cannot be formed as a sum of any subset of the array initially.
2. Iterate through the array, which should be sorted in ascending order. For each element `elem`, check if it is greater than `sum`.
3. If `elem` is greater than `sum`, then `sum` is the smallest number that cannot be formed as a subset sum, and we return it.
4. Otherwise, add `elem` to `sum`, meaning we can now form all sums up to the new `sum`.
5. Continue the process until all elements are processed.

Complexity

Time Complexity:

- The subsequent iteration through the array takes `O(n)`.
- Therefore, the total time complexity is **O(n)**.

Space Complexity:

- The algorithm uses constant extra space apart from the input array, so the space complexity is **O(1)**.

Code

```
class Solution {
  public:
    long long findSmallest(vector<int> &arr) {
            // take input vector as sorted
```

```
        long long sum = 1;

        for (auto elem : arr) {
            if (elem > sum) break;
            sum += elem;
        }

        return sum;
    }
};
```