2.You are given a 0-indexed integer array coins, representing the values of the coins available, and an integer target. An integer x is obtainable if there exists a subsequence of coins that sums to x. Return the minimum number of coins of any value that need to be added to the array so that every integer in the range [1, target] is obtainable. A subsequence of an array is a new non-empty array that is formed from the original array by deleting some (possibly none) of the elements without disturbing the relative positions of the remaining elements.

Example 1:

Input: coins = [1,4,10], target = 19

Output: 2

Program:

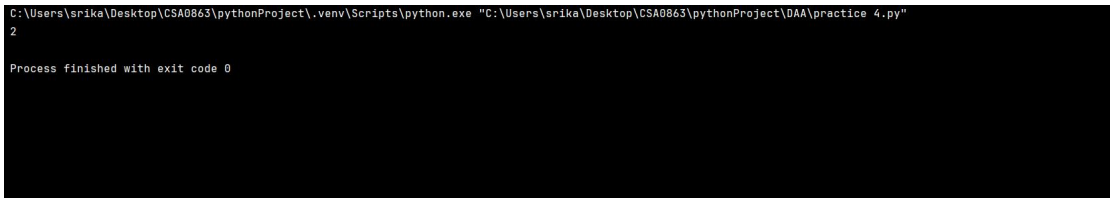
```
def min_coins_to_obtain_target(coins, target):
    coins.sort()
    current_max_reachable = 0
    added_coins_count = 0
    for coin in coins:
        while current_max_reachable + 1 < coin:
            # Add a coin to cover the gap
            added_coins_count += 1
            current_max_reachable += current_max_reachable + 1
            if current_max_reachable >= target:
                return added_coins_count
        current_max_reachable += coin
        if current_max_reachable >= target:
            return added_coins_count
    while current_max_reachable < target:
        added_coins_count += 1
        current_max_reachable += current_max_reachable + 1
    return added_coins_count
```

coins = [1, 4, 10]

target = 19

print(min_coins_to_obtain_target(coins, target))

Output:



Time complexity:

O(nlogn)