5.2 MINIMUM COINS TO MAKE ALL SUMS UP TO TARGET

Question:

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.

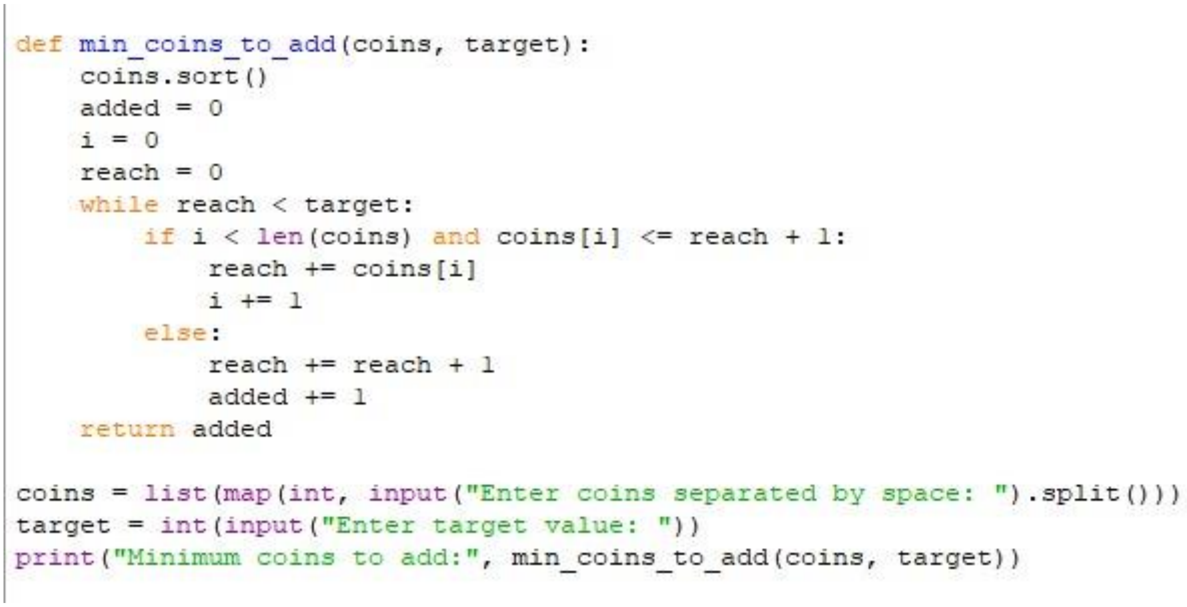
AIM

To determine the minimum number of coins to add so that all integers from 1 to target can be formed using subsequences of the coin array.

ALGORITHM

1. Sort the coins array.
2. Initialize miss = 1, which represents the smallest sum that cannot yet be formed.
3. Initialize i = 0 and added = 0.
4. While miss ≤ target:
   - If i < len(coins) and coins[i] ≤ miss:
     - Add coins[i] to the reachable range: miss += coins[i]
     - Move to next coin: i += 1
   - Else:
     - Add a new coin of value miss to cover the gap
     - Update miss += miss
     - Increment added += 1
5. Return added.

PROGRAM



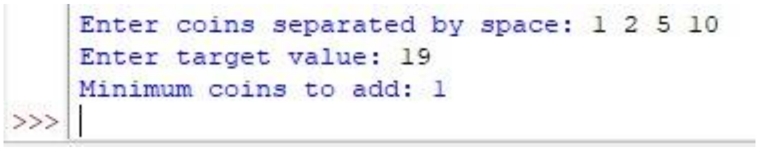
```python
def min_coins_to_add(coins, target):
    coins.sort()
    added = 0
    i = 0
    reach = 0
    while reach < target:
        if i < len(coins) and coins[i] <= reach + 1:
            reach += coins[i]
            i += 1
        else:
            reach += reach + 1
            added += 1
    return added

coins = list(map(int, input("Enter coins separated by space: ").split()))
target = int(input("Enter target value: "))
print("Minimum coins to add:", min_coins_to_add(coins, target))
```

Input:

Enter coins separated by space: 1 2 5 10

Enter target value: 19

Output:



```
Enter coins separated by space: 1 2 5 10
Enter target value: 19
Minimum coins to add: 1
>>>
```

RESULT:

Thus the program is successfully executed and the output is verified.

PERFORMANCE ANALYSIS:

- Time Complexity:
  - O(n log n + target)
- Space complexity:
  - O(1)