

## Medium

An integer  $x$  is obtainable if there exists a subsequence of coins that sums to  $x$ .

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.

```
1 <= target <= 105
1 <= coins.length <= 105
1 <= coins[i] <= target
```

Explanation: We need to add coins 2 and 8. The resulting array will be [1,2,4,8,10]. It can be shown that all integers from 1 to 19 are obtainable from the resulting array, and that 2 is the minimum number of coins that need to be added to the array

Explanation: We only need to add the coin 2. The resulting array will be [1,2,4,5,7,10,19]. It can be shown that all integers from 1 to 19 are obtainable from the resulting array, and that 1 is the minimum number of coins that need to be added to the array.

Explanation: We need to add coins 4, 8, and 16. The resulting array will be [1,1,1,4,8,16]. It can be shown that all integers from 1 to 20 are obtainable from the resulting array, and that 3 is the minimum number of coins that need to be added to the array.

Diagram illustrating the execution of the 'if' and 'elif' branches in the provided code. The diagram shows the state of the 'summation' and 'count' arrays after processing the value 7. The 'summation' array contains [1, 3, 7, ..., 15, 25, ...] and the 'count' array contains [0, 1, ..., 2, ...]. The 'if' branch (if i in amounts: summation += amounts[i]) is shown as the first condition, and the 'elif' branch (elif i > summation: count += 1; summation += i) is shown as the second condition. Arrows indicate the flow of execution from the 'i' variable to the 'if' and 'elif' blocks, and from the 'count' array to the 'return count' statement.