Dynamic Programming

Four steps to success:

1. Identify recursive solution w/overlapped subproblems
2. Use memoization (cache results for reuse)
3. Use iteration instead of recursion
4. Reduce cache size

(Ex Fibonacci calculation (nth number in sequence)

0 1 1 2 3 5 ...

(Ex "How many" calculation

Elements 1 2 3 5 8

Sum 14

↓

answers * 1 5 8

1 2 3 8

2 2 2 8

1 1 2 2 5 5

etc

Recursive
Solution

$N = \min\{\#(sum-1), \#(sum-2), \#(sum-3),$
$\#(sum-5), \#(sum-8)\}$?

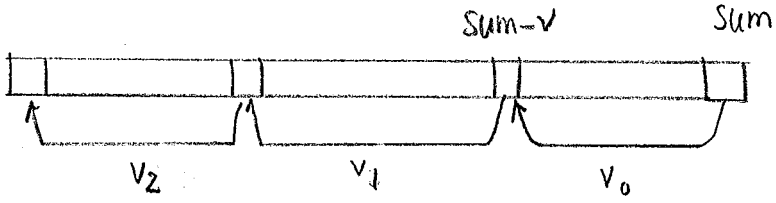
Lots of overlap

Add code for printing solution : link, map ⇒ show

V1 : Sequence of elements used

V2 : vcount array $O(N^2)$

V3 : vcount map $O(N \log N)$



link ≡ currentsum − value of "coin"

⇓ value of "coin" = currentsum − link