

* Sum of Subset

Given n distinct positive number, find all combinations of these numbers whose sum are m .

Solⁿ in n tuple, index \rightarrow index of number in (weights) $w[j]$
value \rightarrow 1 or 0, 1 for considering that weight else not.

Assumption $w[1] \leq m$ and $\sum_{i=1}^n w[i] \geq m$

Explanation

Two conditions \rightarrow for each weight in array

i) case-1 \Rightarrow that value is considered for sum

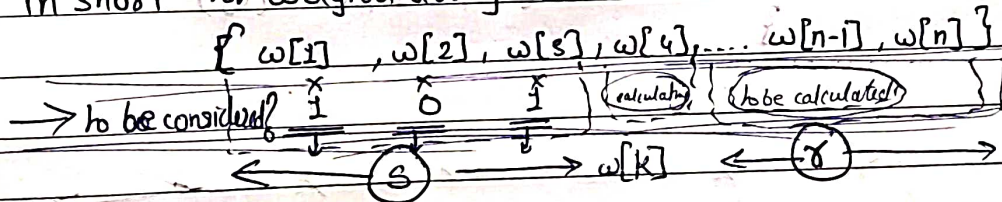
ii) case-2 \Rightarrow not considered

let $s \Rightarrow$ current sum $\left\{ s = \sum_{j=1}^{k-1} w[j] * x[j] \right\}$

$k \Rightarrow$ current index

$r \Rightarrow$ total value left $\left\{ r = \sum_{j=k}^n w[j] \right\}$

in short for weights array w



Steps

① fⁿ SumOfSub(s, k, r)

initially $s=0, k=0, r = \sum_{j=1}^n w[j]$

from 2 conditions

1) #Left child, considering the value hence ($x[k]=1$)

• $x[k]=1$

• if previous sum (s) + this value ($w[k]$) = m , we got answer

so print this
if not above then
• if previous sum + $w[k]$ + next value $w[k+1] \leq m$, ie still does not reach m , we add current sum with $w[k]$ and remove $w[k]$ from total left value (r)

2) #Right child ($x[k]=0$)

• $x[k]=0$

• don't add $w[k]$ in s but remove $w[k]$ from r .

$x[...]$ \leftarrow 1's & 0's
 $w[...]$ \leftarrow values of weights

SumOfSub(s, k, r)

left child

{
 $x[k] \leftarrow 1$
 if ($s + w[k] = m$) then write ($x[1:n]$)
 else if ($s + w[k] + w[k+1] \leq m$)
 then SumOfSub($s + w[k], k+1, r - w[k]$)

if I include $w[k]$, it should not happen that $w[k+1]$ leads to value greater than m , if it does don't add to s

right child

if ($(s + r - w[k] \geq m)$ and
 $(s + w[k+1] \leq m)$)
 then do {
 $x[k] \leftarrow 0$
 SumOfSub($s, k+1, r - w[k]$)

① for ($s + r - w[k] \geq m$)

If I ignore $w[k]$, then the remaining elements in array w , must atleast be able to reach a sum of m (greater than is also acceptable)

② for ($s + w[k+1] \leq m$)

If I ignore $w[k]$, then the next element should not lead to sum being greater than m .