Algorithm SumOfSub(s, k,r)

// Find all subsets of w[1: n] that sum to m. The values of x[j],

//1 <= j < k have already been determined. s =

// and r = The w[j]'s are in nondecreasing order.

// It is assumed that w[1] <= m and >= m .

{

// Generate left child. Note: s + w[k] < m since Bk-1 is true.

x[k] := 1;

if (s + w[k] = m) then write (x[1 / k]) ; // Subset found

// There is no recursive call here as w[j] > 0 1 <= j <= n .

else if (s + w[k] + w[k + 1] <= m)

then SumOfSub (s + w[k], k + 1, r - w[k]);

// Generate right child and evaluate Bk.

if ((s+r-w[k] ≥ m) and (s + w[k + 1] <= m) ) then

{

x[k] := 0;

SumOfSub (s, k + 1, r - w[k]);

}

}