Sum of Subsets.

Subset Sum problem is to find subset of elements that are selected from a given set whose sum adds up to a given number K. We are considering the set contains non-negative values. It is assumed that the input set is unique (no duplicates are presented).

Consider the following example.

$$w[1:6] = \{ \overset{1}{5}, \overset{2}{10}, \overset{3}{12}, \overset{4}{13}, \overset{5}{15}, \overset{6}{18} \} \qquad n = 6.$$

We have to take subsets of those weights such that their sum total is exactly equal to a given value (here it is 30).

ie, $m = 30$

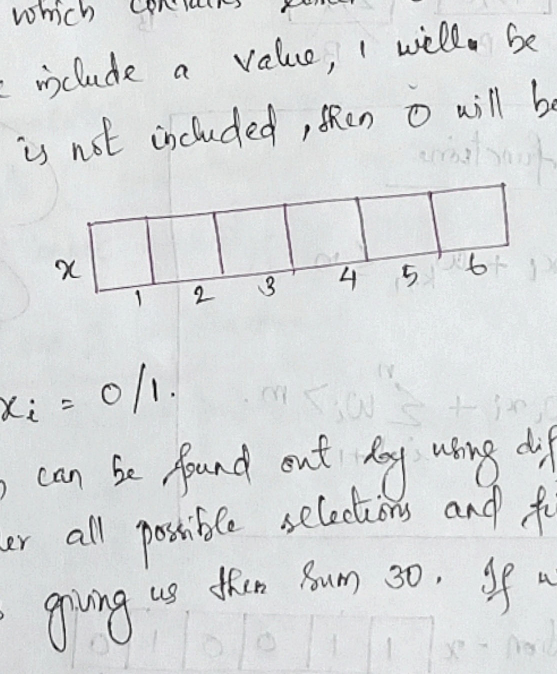
Which all weights are including; we will write the solution in an array which contains either 0 or 1 as values.

ie; If we include a value, 1 will be written the array and if a value is not included, then 0 will be written.

x						
	1	2	3	4	5	6

$x_i = 0/1.$

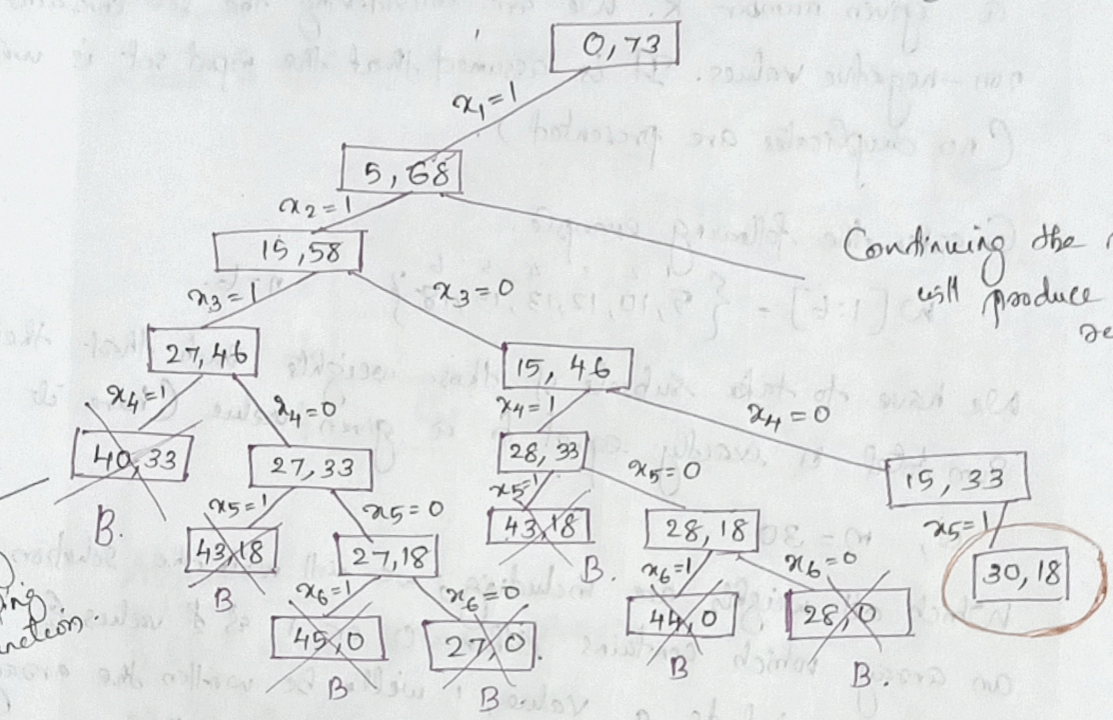
The solution can be found out by using different methods. One way is to consider all possible selections and find out which of those selections is giving us the sum 30. If we draw the state space tree,



But it extremely time consuming and the complexity is 2^n.

If we apply backtracking we have to kill the nodes by applying bounding function. The state space tree using backtracking is

$w[1:6] = \{5, 10, 12, 13, 15, 18\}$



Continuing the process will produce more results.

By applying bounding function

Bounding function.

$$\sum_{i=1}^{k} w_i x_i + w_{k+1} \leq m$$

$$\sum_{i=1}^{k} w_i x_i + \sum_{i=k+1}^{n} w_i > m.$$

$$SoS(n, m) = \begin{cases} SoS(n-1, m) \\ \text{OR} \\ SoS(n-1, m-w_n) \end{cases}$$

n = Remaining weight
m = Sum of n weights

Solution - x	1	1	0	0	1	0
	1	2	3	4	5	6

Complexity = $\underline{O(nm)}$