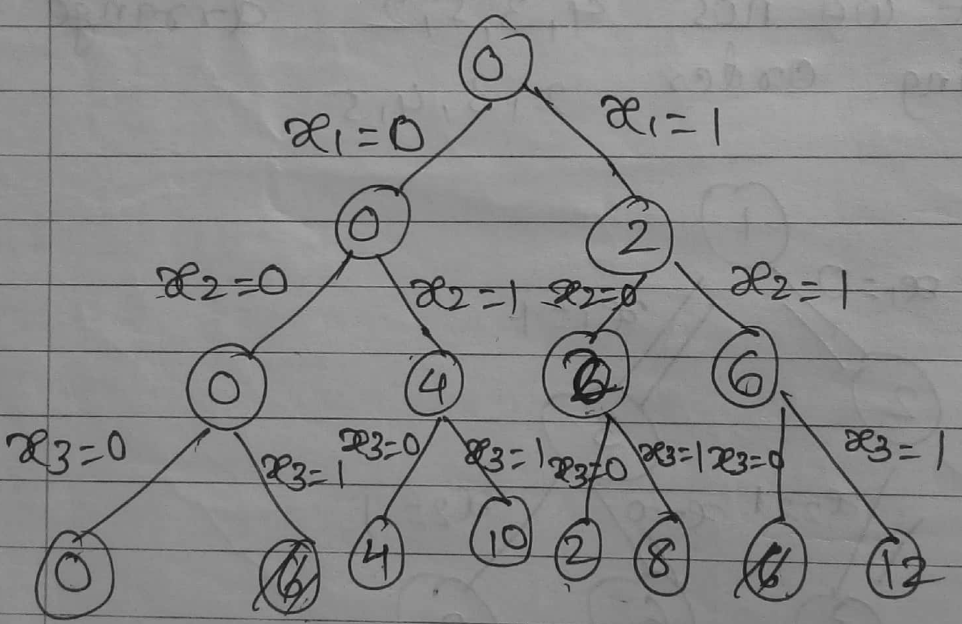
Sum of subsets

n +ve distinct nos are given they are referred as weights $w_1, w_2, --- w_n$ & variable S is given. We have to find such a combination of weights whose sum = S.

Sol^n is represented by binary state space tree and one array in which $x_i = 1$ if w_i is considered

$S = 6$, $w_1 = 2$, $w_2 = 4$, $w_3 = 6$



Sol^n vector is [0 0 1]

[1 1 0]

(2) $S = 13$

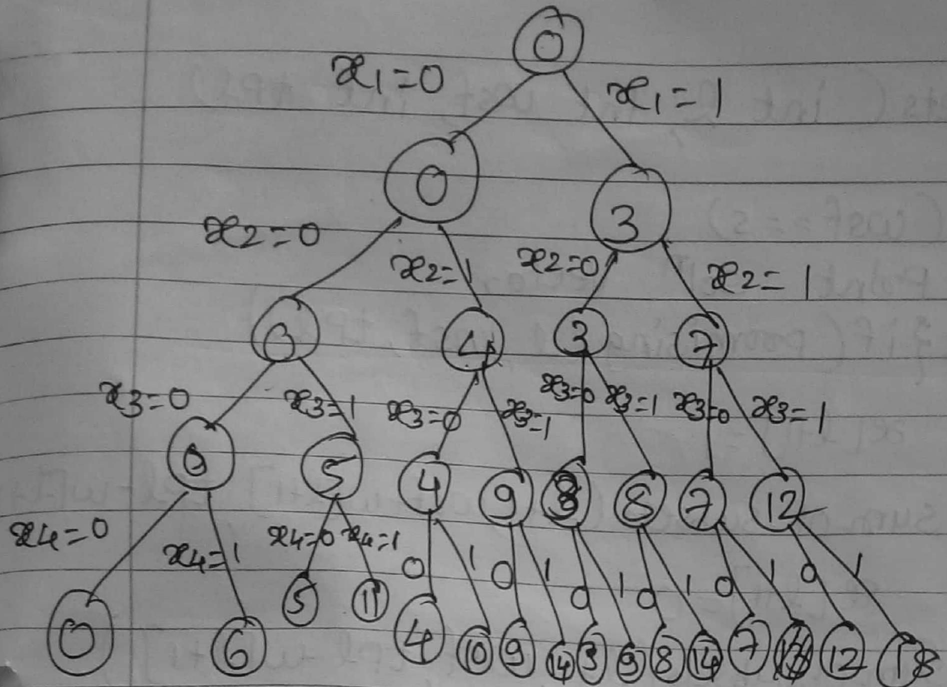
w 3 4 5 6

$3+$

$wsf + tPL >= S$

$4 + 6 >= 13$

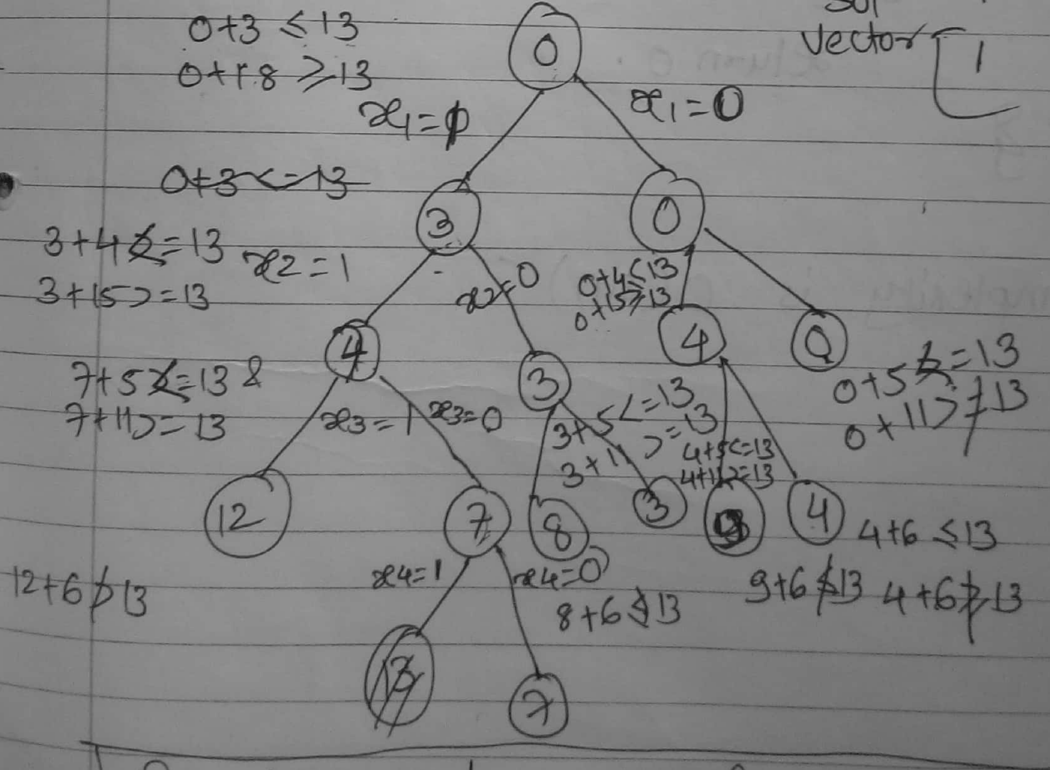
[1 1 0 1]



It is not feasible to expand all nodes as this method is not efficient.

The efficient method is to expand a node only if it is promising that is only if it can gives you a solution.

Sol^n vector	1	2	3	4
	1	1	0	1



$$\boxed{\text{Promising node} \quad wsf + tPL >= S \;\&\&\; wsf + w[l+1] <= S}$$

```
sum_of_subsets ( int l, int wsf, int tpl)
 { if (wsf == s)
      Print soln vector
   else { if( promising (l, wsf, tpl))
     { x[l+1] = 1
       sum_of_subsets (l+1, wsf+w[l+1], tpl-w[l+1])
         x[l+1] = 0
       sum_of_subsets ( l+1, wsf, tpl-w[l+1] )
     }
 }

int  promising (int l, int wsf, int tpl)
 { if( wsf+ w[l+1] <= s && wsf+tpl >= s)
          ~~retur~~ return 1;
     else
          return 0;
 }
```

complexity is O(s.n)