Sum Of Subsets Problem

Suppose we are given n distinct +ve no's (usually called weights) and we desire to find all combinations of these no's whose sums are m. This is called the sum of subsets problem.

Given +ve no's w_i, $1 \le i \le n$, and m, this problem calls for finding all subsets of the w_i whose sums are m. For example, if $n=4$, $(w_1 w_2 w_3 w_4) = (11, 13, 24, 7)$ and $m=31$, then the desired subsets are $(11, 13, 7)$, $(24, 7)$. Rather than representing the solution vector by the w_i which sum to m, we could represent the solution vectors by giving the indices of these w_i. Now the two solutions are described by the vectors $(1,2,4)$ and $(3,4)$. In general, all solutions are k-tuples $(x_1, x_2, \ldots x_k)$, $1 \le k \le n$ and different solutions may have different sized tuples. The explicit constraint requires $x_i \in \{ j \mid j$ is an integer and $1 \le j \le n \}$. The implicit constraints require that no two be the same and that the ~~same~~ sum of the corresponding w_i's be m. Since we wish to avoid generating multiple instances of the same subset (e.g. $(1,2,4)$ and $(1,4,2)$ represents same subset)

Another implicit constraints that is impo[sed] is that $x_i < x_{i+1}$, $1 \le i \le k$.

In another formulation of the sum of subsets problem, each solution subset is represented by an n-tuple $(x_1, x_2, x_3 \ldots x_n)$ such that $x_i \in \{0,1\}$, $1 \le i \le n$. Then $x_i = 0$ if w_i is not chosen and $x_i = 1$ if w_i is chosen. The solutions to the above instance are $(1,1,0,1)$ & $(0,0,1,1)$. This solution expresses all solutions using <u>fixed-sized</u> tuple

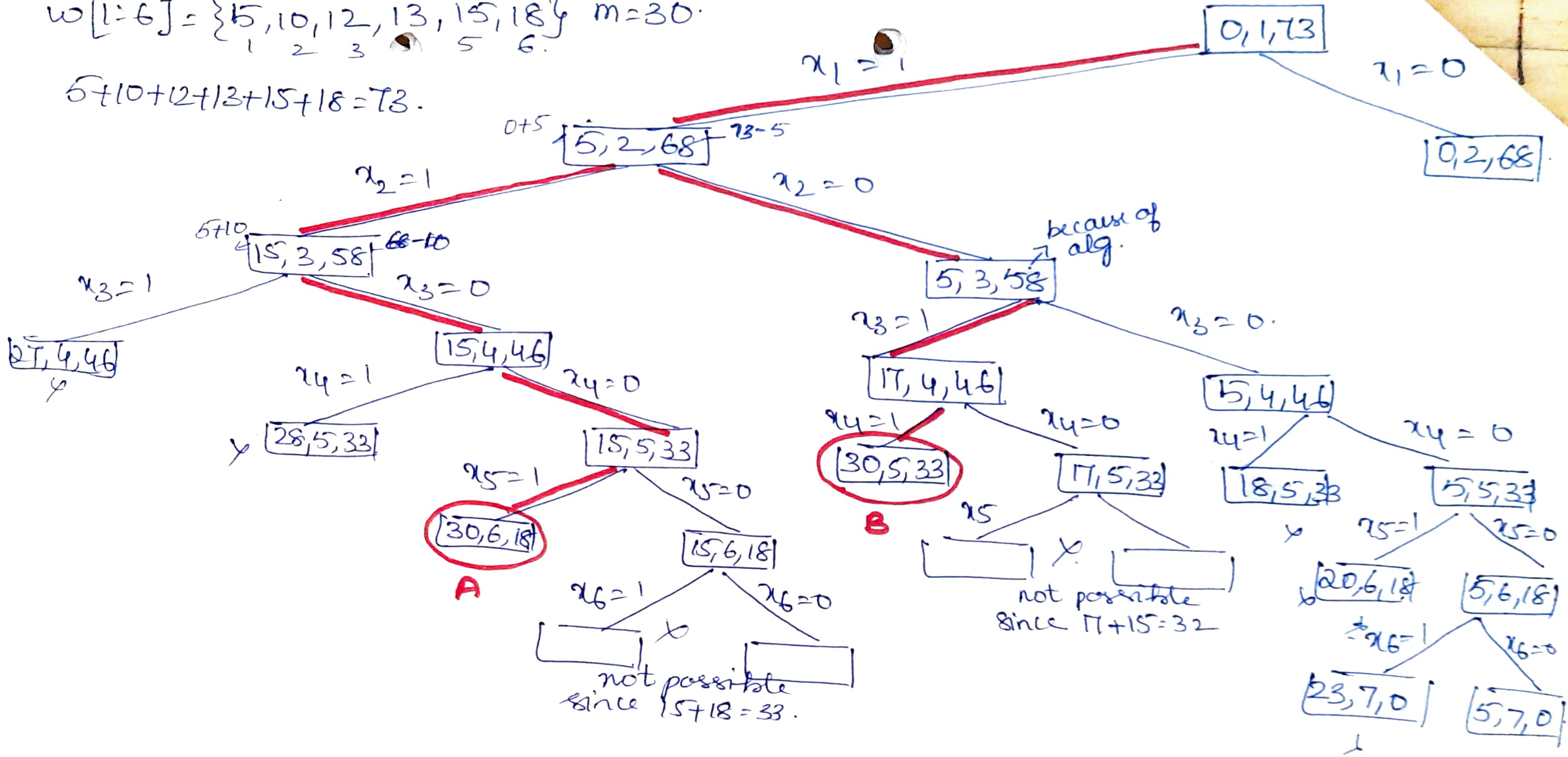
The solution can be represented using state space tree diagram. In this tree the children of any node are easily generated. For a node at level i the left child corresponds to $x_i = 1$ and the right to $x_i = 0$.

<u>Problem:</u>

Show the portion of the state space tree generated by the function SumOfSub. while working on the instance $n=6$, $m=30$, and $w[1:6] = \{5, 10, 12, 13, 15, 18\}$.

$w[1:6] = \{5, 10, 12, 13, 15, 18\}$ $m = 30$.

$5+10+12+13+15+18 = 73$.



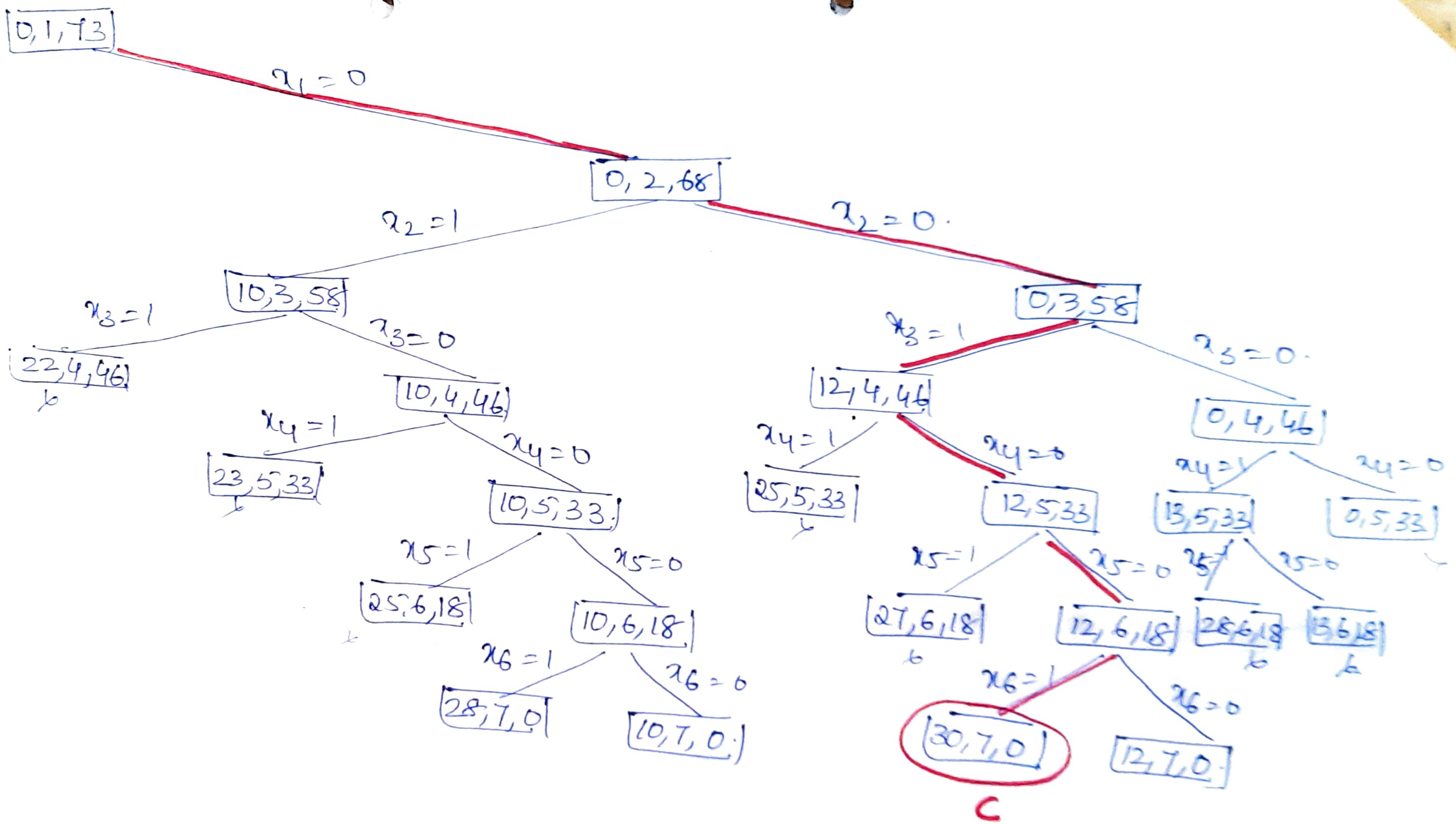
3 subsets are possible which are indicated by A, B, C.

Solutions vectors are

A = (1, 1, 0, 0, 1) $5+10+15 = 30$.

B = (1, 0, 1, 1) $5+12+13 = 30$

C = (0, 0, 1, 0, 0, 1) $12+18 = 30$.



0, 1, 73
x1 = 0
0, 2, 68
x2 = 1
x2 = 0
10, 3, 58
0, 3, 58
x3 = 1
x3 = 0
22, 4, 46
10, 4, 46
12, 4, 46
0, 4, 46
x4 = 1
x4 = 0
23, 5, 33
10, 5, 33
25, 5, 33
12, 5, 33
13, 5, 33
0, 5, 33
x5 = 1
x5 = 0
25, 6, 18
10, 6, 18
27, 6, 18
12, 6, 18
28, 6, 18
13, 6, 18
x6 = 1
x6 = 0
28, 7, 0
10, 7, 0
30, 7, 0
12, 7, 0
C

Algorithm SumOfSub (s, k, r)

// Find all subsets of $w[1:n]$ that sum to m.

// The values of $x[j]$, $1 \le j \le k$, have already

// been determined. $s = \sum_{j=1}^{k-1} w[j] * x[j]$ and

// $r = \sum_{j=k}^{n} w[j]$. The $w[j]$'s are in nondecreasing

// order. It is assumed that $w[1] \le m$ and

// $\sum_{i=1}^{n} w[i] \ge m$.

```
{
    // generate left child.
        x[k] := 1;
    if (s + w[k] = m) then ~~write~~
        write (x[1:k]); // Subset found.
    else if (s + w[k] + w[k+1] ≤ m)
        then SumOfSub (s + w[k], k+1, r - w[k]);
    // generate right child.
    if ((s + r - w[k] ≥ m) and (s + w[k+1] ≤ m)) then
    {
        x[k] := 0;
        SumOfSub (s, k+1, r - w[k]);
    }
}
```

The initial call is SumOfSub$(0, 1, \sum_{i=1}^{n} w_i)$.

In our example root node ~~is~~ contains

$$\boxed{0, 1, 73}$$

$$\begin{array}{ccc} \downarrow & \downarrow & \downarrow \\ s & k & r \end{array}$$

s – current weight (add on).

k – level number.

r = total sum of weights.

Initially $k = 1$. (since root node)

$$s = \sum_{j=1}^{k-1} w[j] * x[j] = 0.$$

$$r = \sum_{j=k}^{n} w[j] = 73.$$

The portion of state space tree is obtained by tracing the algorithm.