DESIGN AND ANALYSIS OF ALGORITHMS

Session -27

SUM OF SUBSETS PROBLEM BY BACKTRACKING

SUM OF SUBSETS

Problem Definition: Given n distinct positive numbers wi, and m, find all subsets whose sums are m.

- Explicit Constraints :
- Xi = { j / j is an integer and 1 <= j <= n }
- Implicit Constraints :

1. No two xi’s are same
2. Sum of the chosen weights must be equal to m.
3. To avoid generation of multiple instances of the same subsets)

We can formulate this problem using Fixed tuple or Variable tuples.

VARIABLE- SIZED TUPLE

Ex:- n=4, (w1, w2, w3, w4)= (11,13, 24, 7), m=31.

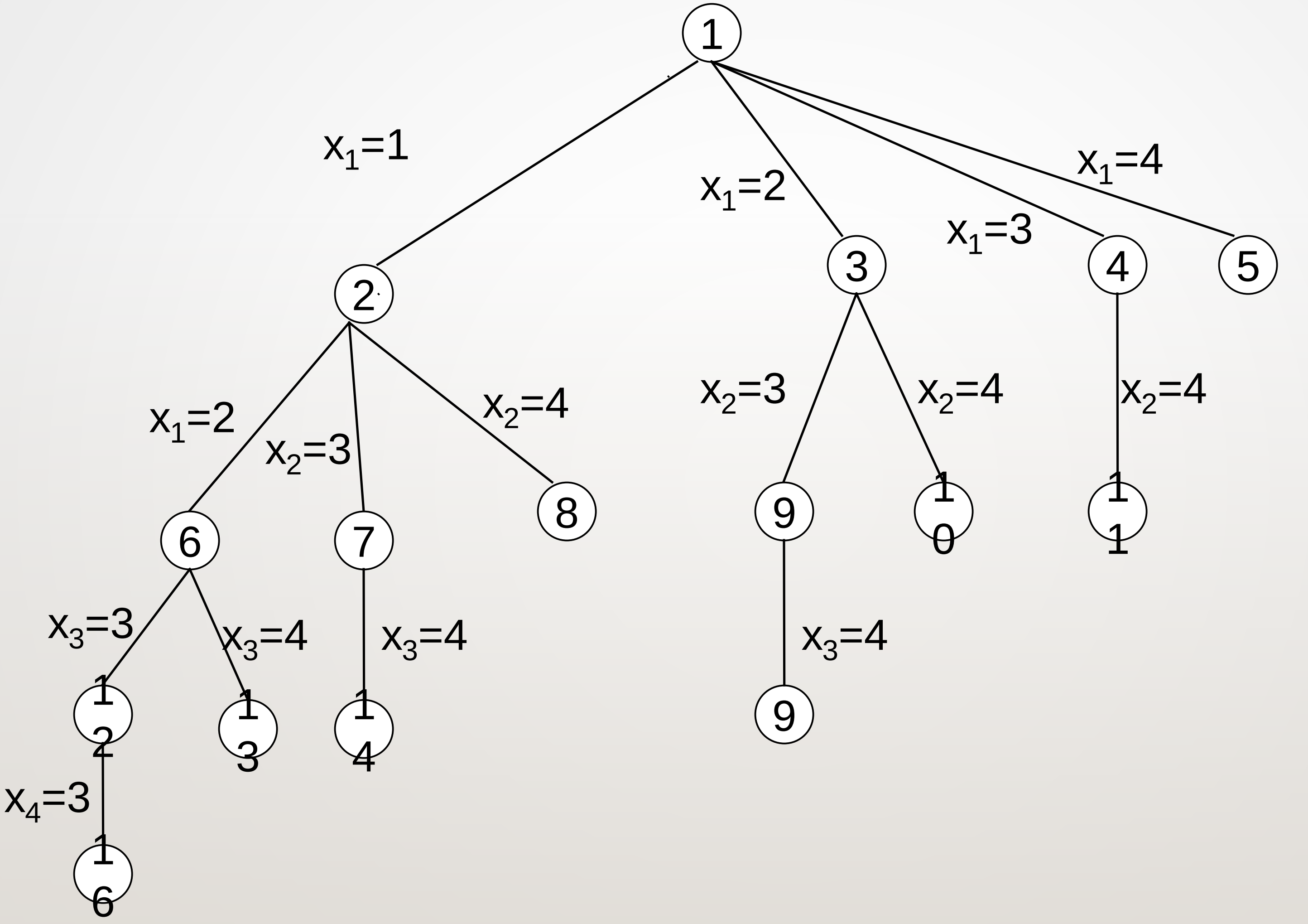
- Solutions are (11, 13, 7) and (24, 7)

- Rather than representing the solution by wi 's, we can represent by giving the indices of these wi
- Now the solutions are (1, 2, 4) and (3, 4).

- Different solutions may have different-sized tuples.
- We use the following condition to avoid generating multiple instances of the same subset (e.g., (1,2,4) and (1,4,2))

$$x_i < x_i+1$$

Subset Tree for n=4 (variable- sized tuple)



Nodes are numbered as in Breadth first search.

FIXED- SIZED TUPLE

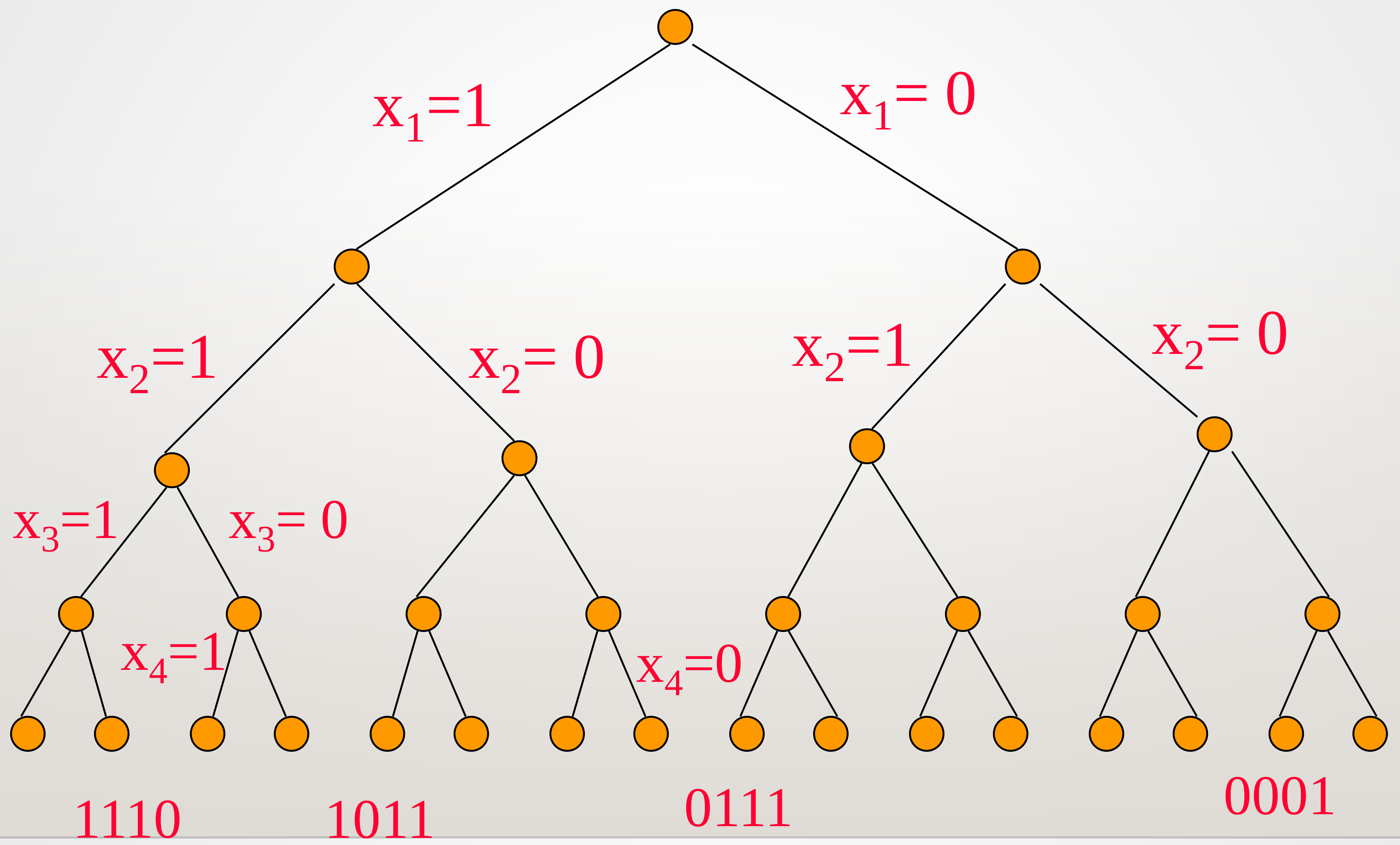
- In this method, each solution subset is represented by an n-tuple (x1,x2,…..x n).
- x i = 0 if wi is not chosen and xi = 1 if wi is chosen

Ex:- n=4, (w1, w2, w3, w4)= (11,13, 24, 7), m=31.

- Solutions are (11, 13, 7) and (24, 7)

- Solutions are:
- X=(1,1,0,1) and X=(0,0,1,1)

SUBSET TREE FOR N = 4 (FIXED – SIZED TUPLE)



BOUNDARY CONDITIONS OF SUM OF SUBSETS

Simple choice for the bounding Function is Bk (X1 … Xk) = true iff

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

Clearly x1 …xk can not lead to an answer node if this condition is not satisfied.

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

Assuming wi’s in non decreasing order, (x1... xk) cannot lead to an answer node if

So, the bounding functions we use are therefore

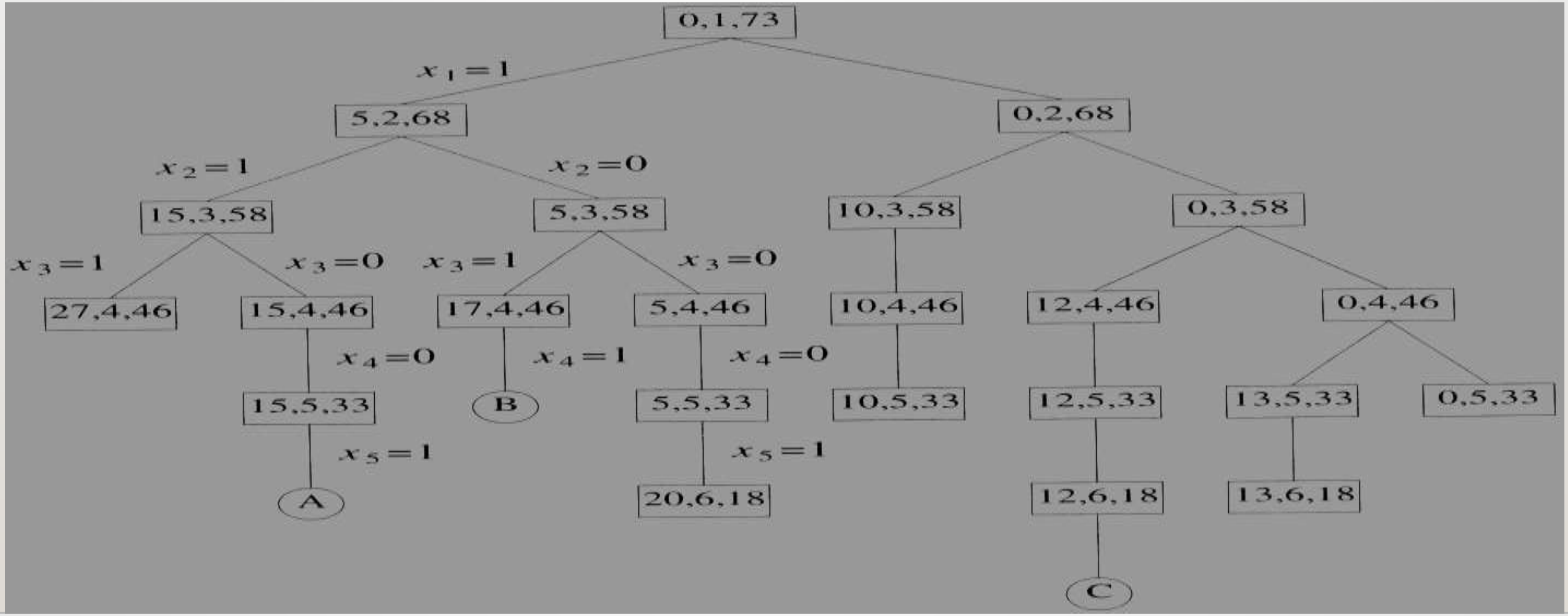
$$B_k(x_1, \ldots, x_k) = true \ iff \sum_{i=1}^{k} w_i x_i + \sum_{i=k+1}^{n} w_i \geq m$$

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

```
Algorithm SumOfSub( s, k, r )
// Find all  subsets of w[ 1:n ]  that sum to m
// It is assumed that w[1] ≤ m  and    ∑     wi ≥  m
{
        x[ k ]=1;  //  left child
        if( s + w[k] = m )  then write( x[ 1: k ] ) ; // Subset found
      else  if ( s + s [ k ] + s[ 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] ) ;
}
```

EX:- N=6, M=30, W [1:6]= { 5,10,12,13,15,18 }
PORTION OF STATE SPACE TREE GENERATED BY SUMOFSUB.
CIRCULAR NODES INDICATE SUBSETS WITH SUMS EQUAL TO M.

- Example 7.6 (Figure 7.10)
- n=6, w[1:6]={5,10,12,13,15,18}, m=30



SAMPLE QUESTIONS

- What is the Sum of Subsets problem
- Describe the problem statement of the Sum of Subsets. What are the inputs and outputs of the problem
- Explain about state space tree in sum of subsets problem
- Solve sum of subsets by using Input: set[] = {4, 16, 5, 23, 12}, sum = 9