III rd solution is **C** -> 0 0 1 0 0 1

- In the state space tree, edges from level 'i' nodes to 'i+1' nodes are labeled with the values of Xi, which is either 0 or 1.
- The left sub tree of the root defines all subsets containing Wi.
- The right subtree of the root defines all subsets, which does not include Wi.

GENERATION OF STATE SPACE TREE:

- Maintain an array X to represent all elements in the set.
- The value of Xi indicates whether the weight Wi is included or not.
- Sum is initialized to 0 i.e., s=0.
- We have to check starting from the first node.
- Assign X(k)<- 1.
- If S+X(k)=M then we print the subset b'coz the sum is the required output.
- If the above condition is not satisfied then we have to check S+X(k)+W(k+1)<=M. If so, we have to generate the left sub tree. It means W(t) can be included so the sum will be incremented and we have to check for the next k.
- After generating the left sub tree we have to generate the right sub tree, for this we have to check S+W(k+1)<=M.B'coz W(k) is omitted and W(k+1) has to be selected.
- Repeat the process and find all the possible combinations of the subset.

Preview from Notesale.co.uk
Page 4 of 16

Algorithm:

```
Algorithm sumofsubset(s,k,r)
{
//generate the 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]); // there is no recursive call here as W[j]>0,1<=j<=n.
Else if (S+W[k]+W[k+1]<=m) then sum of sub (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;
```

```
        }
    }
}
```

Algorithm for Bounding function:

```
Algorithm Bound(cp,cw,k)
// cp→ current profit total.
//cw→ current weight total.
//k→the index of the last removed item.
//m→the knapsack size.

{
    b=cp;
    c=cw;
    for I =- k+1 to n do
  {
          c= c+w[I];
        if (c<m) then b=b+p[I];
            else return b+ (1-(c-m)/W[I]) * P[I];
}
return b;
}
```

Example:

Preview from Notesale.co.uk
Page 14 of 16

M= 6 Wi = 2,3,4 4 2 2

N= 3 Pi = 1,2,5 Pi/Wi (i.e.) 5 2 1

Xi = 1 0 1
The maximum weight is 6

The Maximum profit is (1*5) + (0*2) + (1*1)
→ 5+1
→ 6.

Fp = (-1)

- $1 \leq 3$ & $0+4 \leq 6$
  cw = 4,cp = 5,y(1) =1
  k = k+2

- $2 \leq 3$ but 7>6
  so y(2) = 0

- So bound(5,4,2,6)