0/1 Knapsack Problem

Solved With

Least cost Branch and Bound(LCBB)

by

Dr.Jayashree Katti

Objective Function

Maximize $\text{profit} \ldots\ldots \sum p_i x_i$

Coverted to minimization problem

i.e - $\sum p_i x_i$

Bound Algorithm to calculate C

```
Algorithm Bound(cp ,cw, k)
//cp=current profit; cw=current weight;k=index of last
removed item; m is 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;
}
```

Ubound Algorithm to calculate u

```
Algorithm Uboud(cp ,cw, k ,m)
//cp=current profit; cw=current weight;k=index of last removed item; m is knapsack size;//
{
   b=cp;  c=cw;
    for i=k+1 to n do
      {
         if(c+w[i]<=m) then
             {
              c=c+w[i];b=b-p[i];
           }
       }
return b;
}
```

Example

- n=4
- P=[10,10,12,18]
- W=[2,4,6,9]
- m=15

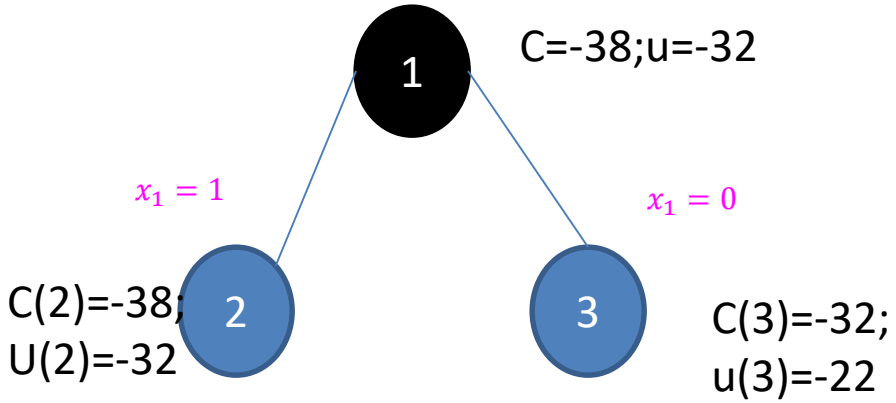
Upper=-32

1

C(1)=-38;
u(1)=-32

List of live nodes	1				

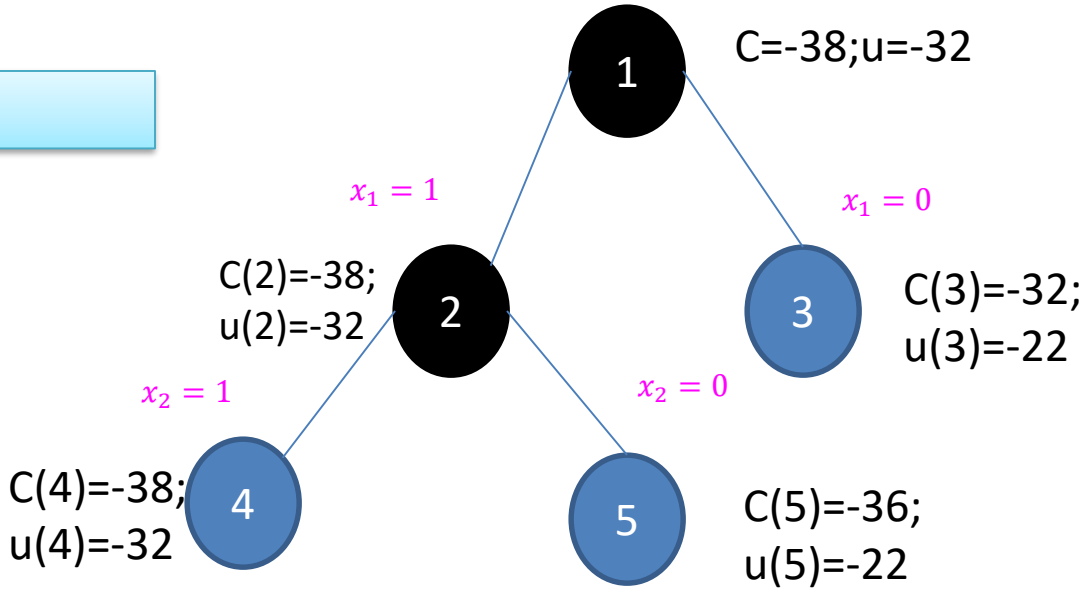
Upper=-32



List of live nodes	2	3			

As c(2)<c(3), Next E-node is 2

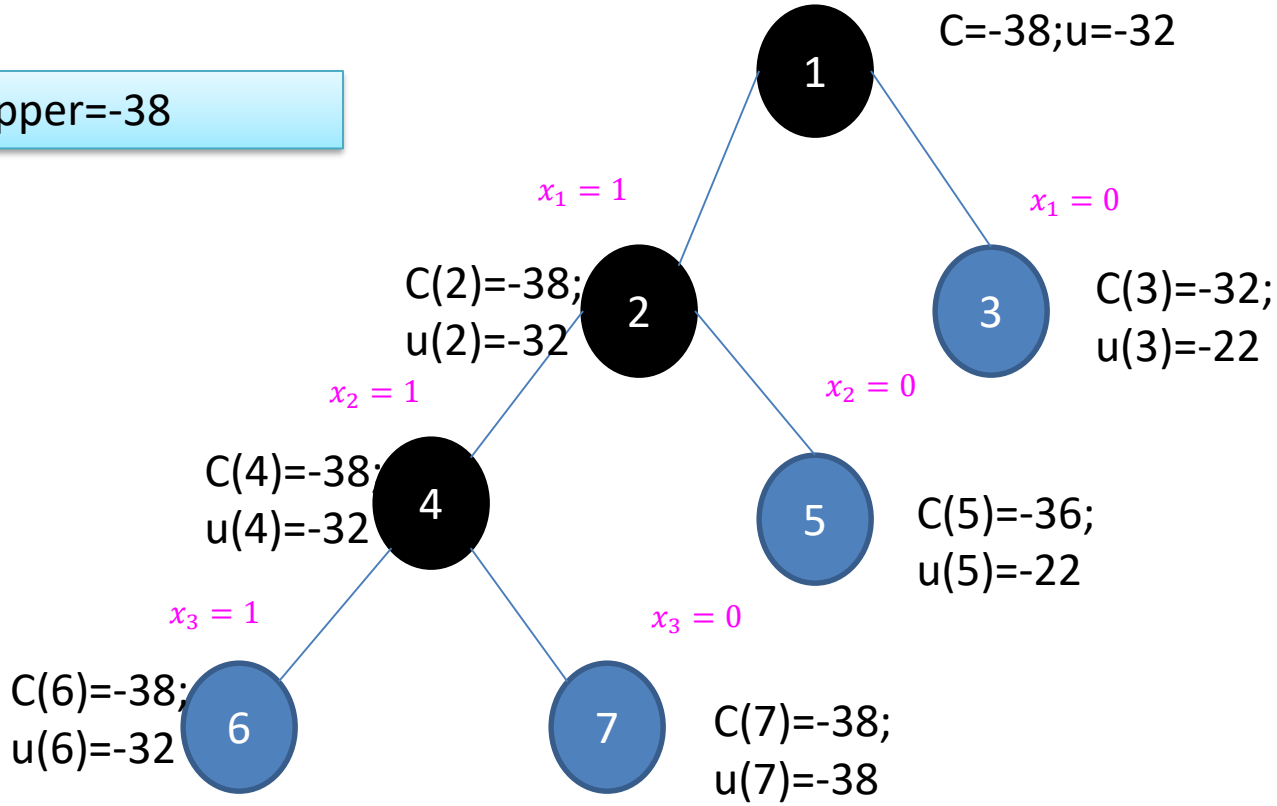
Upper=-32



List of live nodes	3	4	5		

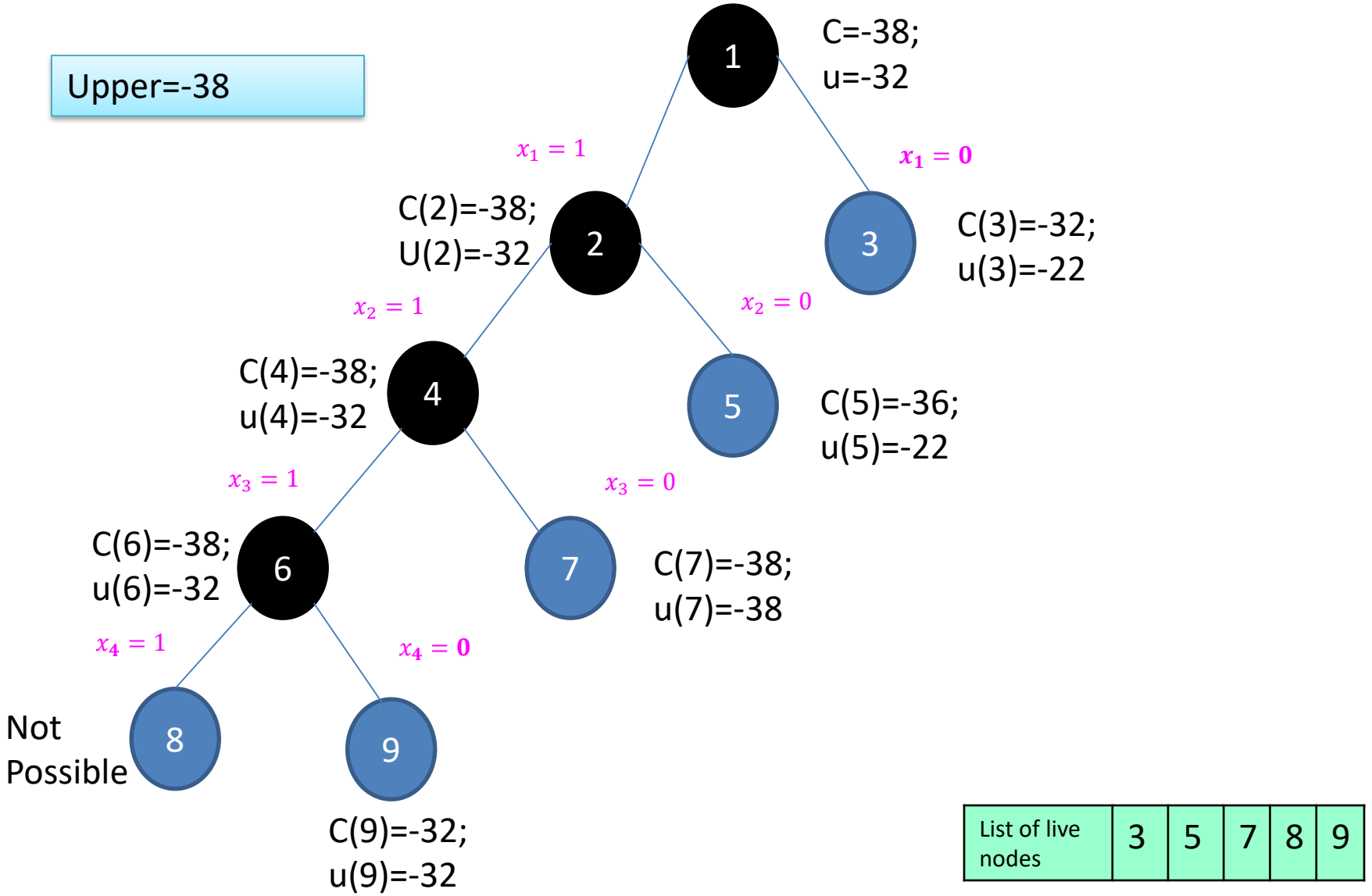
Comparing cost of nodes 3,4,5…
Cost of 4 is minimum. So next E-node is 4

Upper=-38



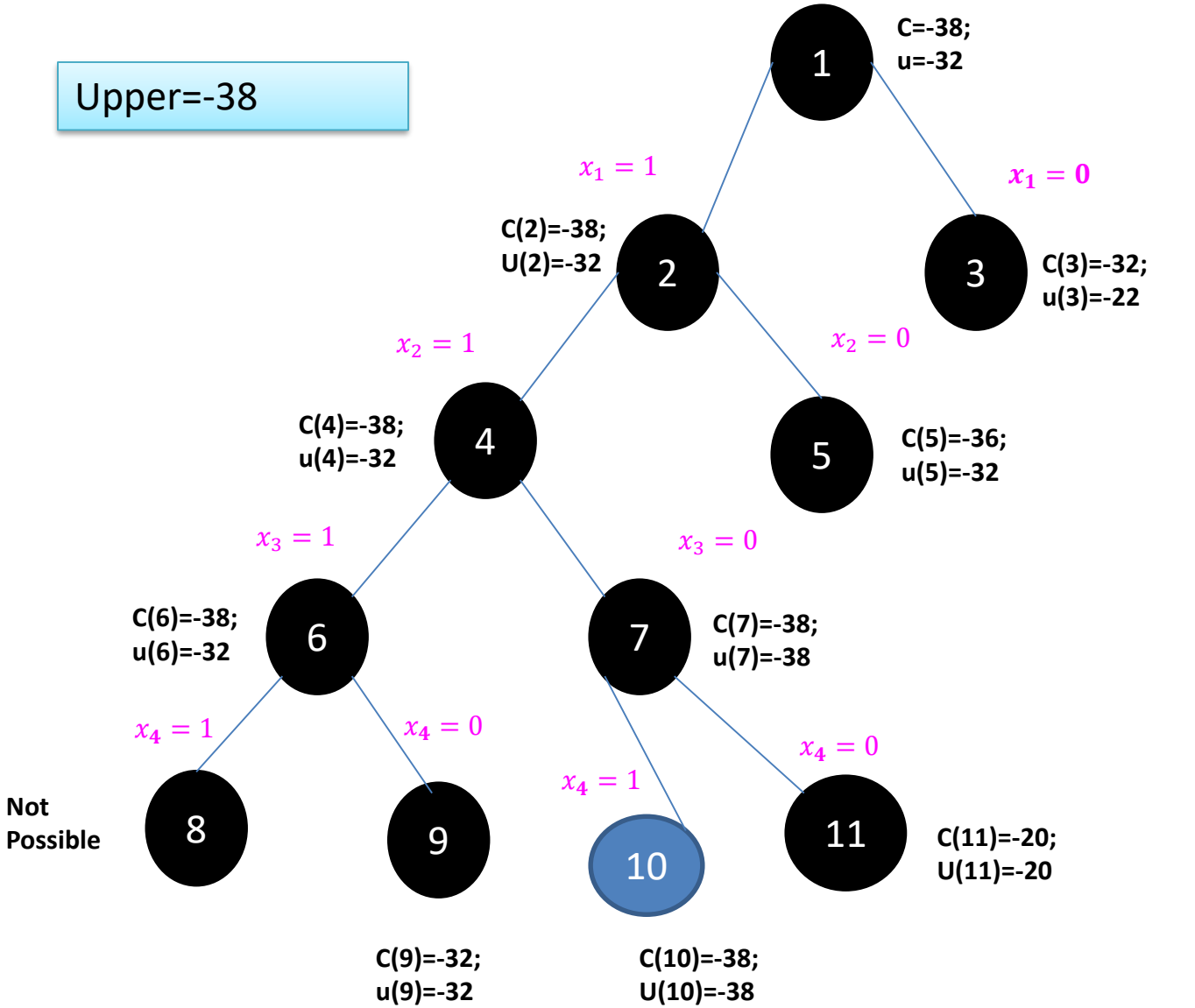
List of live nodes	3	5	6	7	

Comparing cost of nodes 3,5,6,7…
Cost of 6 and 7 are minimum. So next E-node is 6



List of live nodes	3	5	7	8	9

Comparing cost of nodes 3,5,7,8,9
Cost of 7 is minimum. So next E-node is 7



Solution Vector X=[1,1,0,1]