## * 0/1 Knapsack Problem Using Branch & Bound

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Profit ⇒ | 10 | 10 | 12 | 18 |
| weight ⇒ | 2 ✓ | 4 ✓ | 6 ✗ | 9 ✓ |

$M = 15$

$n = 4$

$(0, 0, 1, 1) \Rightarrow P = 30$

Soln ⇒ We know the soln ie ⇒ $(1, 1, 0, 1)$ where Profit gained $= 38$, Wt $= 15$

0/S

- Using B & B we need to obtain State space tree (BFS)
- At every node of state space tree we need to obtain 2 terms

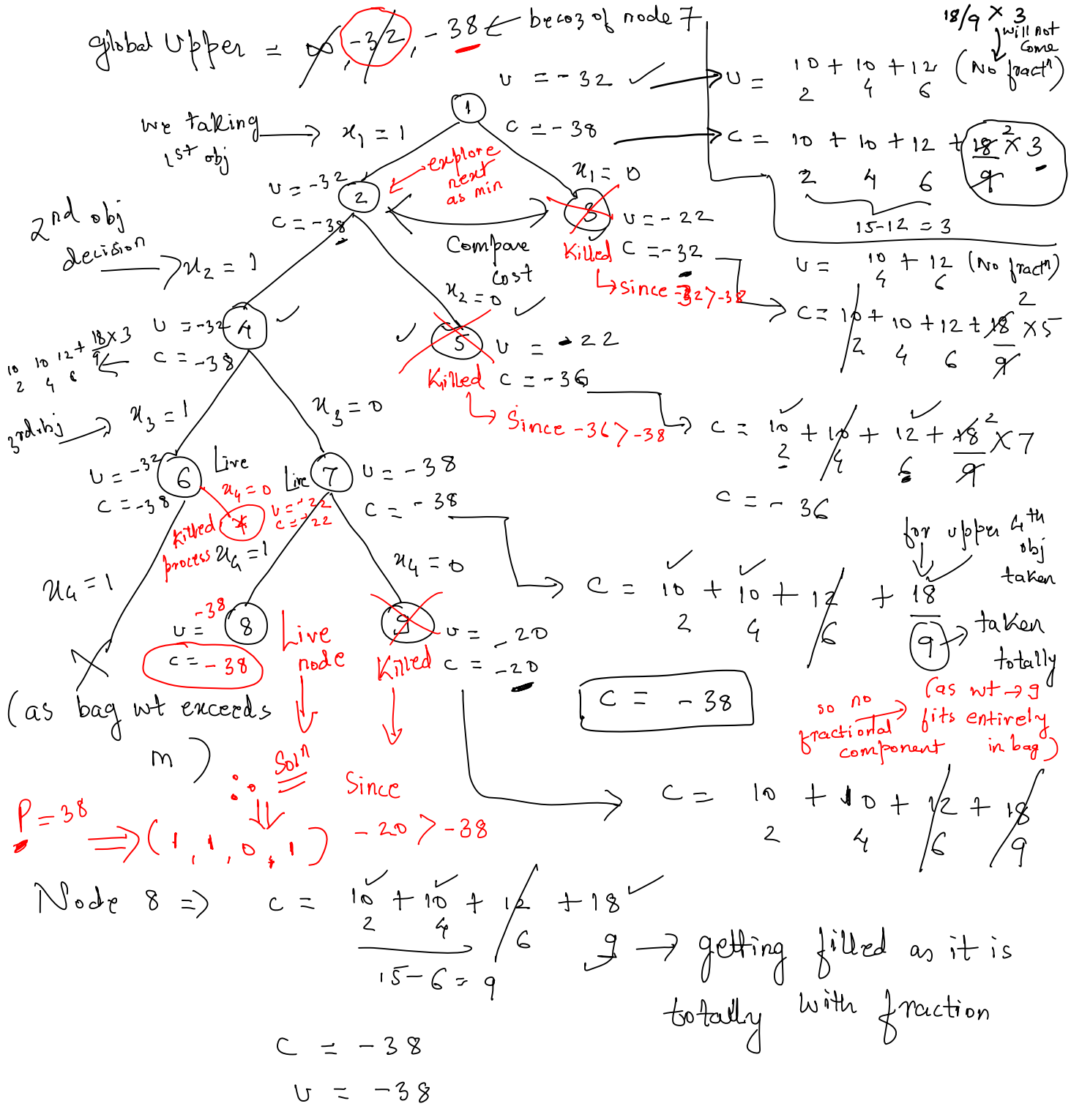
take as one

1st compute

① upper bound $(U) = \sum_{i=1}^{n} P_i x_i \leq m$ → without fraction

② Cost of node $(C) = \sum_{i=1}^{n} P_i x_i$ → with fraction

- Initially we assume upper $= \infty$ (global upper value)
- At any node if the upper value of node (ie U) is less than initial upper assumed (ie global upper value) than update the global upper value
- At any node if cost (ie C) of node is greater than global upper value than kill that node (delete) → if equal than retain node
- Node with least cost is always explored in the process (as it is LC B & B)


global Upper = ∞, -32, -38 ← becoz of node 7
1
u = -32
c = -38
We taking 1st obj → x1 = 1
x1 = 0
2
u = -32
c = -38
explore next as min
3
u = -22
Killed c = -32
↳ since -32 > -38
Compare cost
2nd obj decision → x2 = 1
x2 = 0
4
u = -32
c = -38
5
u = -22
Killed c = -36
↳ Since -36 > -38
3rd obj → x3 = 1
x3 = 0
6
u = -32
c = -38
Live
7
u = -38
c = -38
Live
x4 = 0
u = -22
c = -22
Killed process x4 = 1
x4 = 1
x4 = 0
8
u = -38
c = -38
9
u = -20
Killed c = -20
Live node
(as bag wt exceeds m)
Soln
Since -20 > -38
P = 38
⇒ (1, 1, 0, 1)
Node 8 ⇒ c = 10/2 + 10/4 + 12/6 + 18 ✓
15 - 6 = 9
9 → getting filled as it is totally with fraction
c = -38
u = -38
u = 10/2 + 10/4 + 12/6 (No fract^n)
18/9 × 3 will not come
c = 10/2 + 10/4 + 12/6 + 18/9 × 3
15 - 12 = 3
u = 10/2 + 12/4 + 10/6 (No fract^n)
c = 10/2 + 10/4 + 12/6 + 18/9 × 5
c = 10/2 + 10/4 + 12/6 + 18/9 × 7
c = -36
c = 10/2 + 10/4 + 12/6 + 18
for upper 4th obj taken
9 → taken totally (as wt → 9 so no fractional component fits entirely in bag)
c = -38
c = 10/2 + 10/4 + 12/6 + 18/9