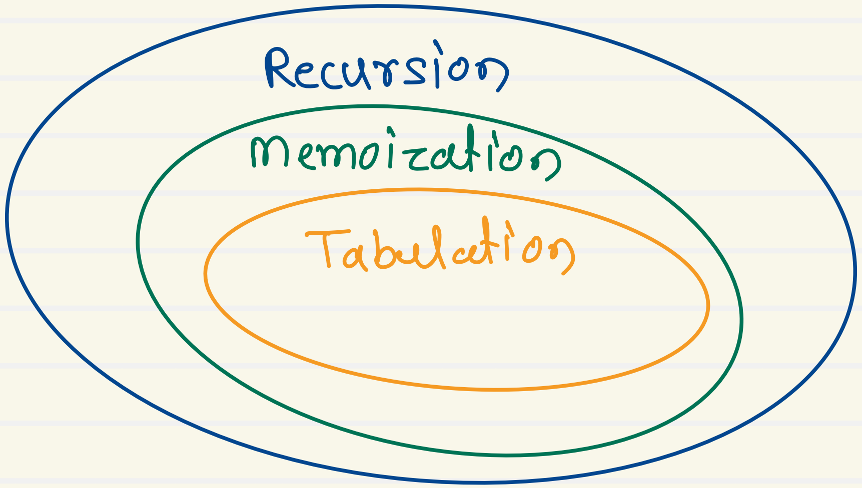
Sunbeam Institute of Information Technology
Pune and Karad

Data structures and Algorithms

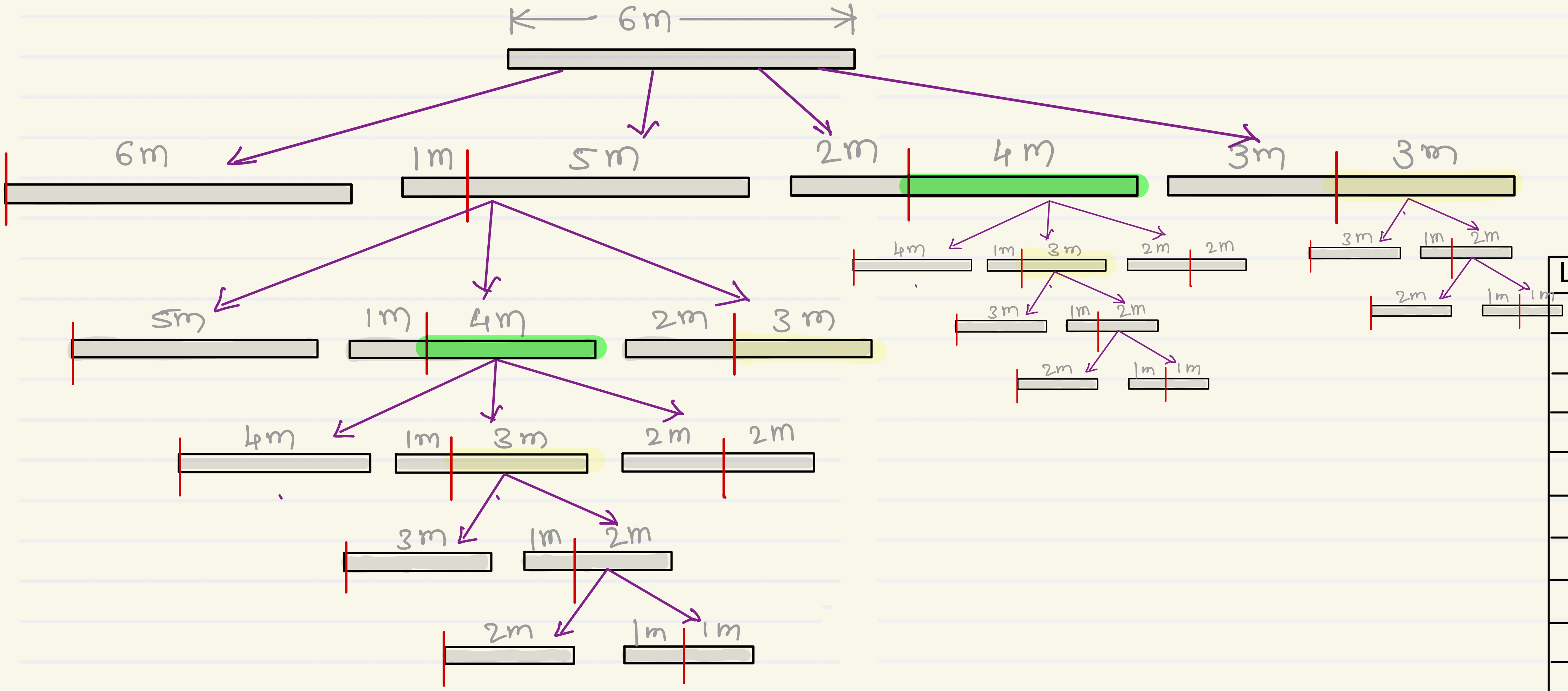
Trainer - Devendra Dhande
Email – devendra.dhande@sunbeaminfo.com

Dynamic Programming

- Memoization
  - Recursion + 1D or 2D Arrays
  - (Top Down approach)
- Tabulation
  - loops + 1D or 2D Arrays
  - (Bottom Up approach)



- Cut the rod of given price so that maximum profit can be achieved.



Length	Price
1	1
2	5
3	8
4	9
5	10
6	14
7	17
8	20
9	24
10	30

Dynamic programming - Rod cutting problem

- Cut the rod of given price so that maximum profit can be achieved.

Length	Price
1	1
2	5
3	8
4	9
5	10
6	14
7	17
8	20
9	24
10	30

rod(1) = 1

rod(2) = 5
- rod(2) = 5
- rod(1) + rod(1) = 2

rod(3) = 8
- rod(3) = 8
- rod(1) + rod(2) = 6

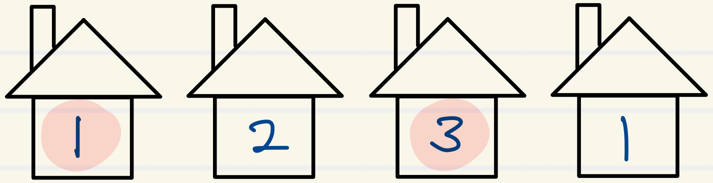
rod(4) = 10
- rod(4) = 9
- rod(1) + rod(3) = 9
- rod(2) + rod(2) = 10

rod(5) = 13
- rod(5) = 10
- rod(1) + rod(4) = 11
- rod(2) + rod(3) = 13

rod(6) = 16
- rod(6) = 14
- rod(1) + rod(5) = 14
- rod(2) + rod(4) = 15
- rod(3) + rod(3) = 16

Dynamic programming - House robbing problem

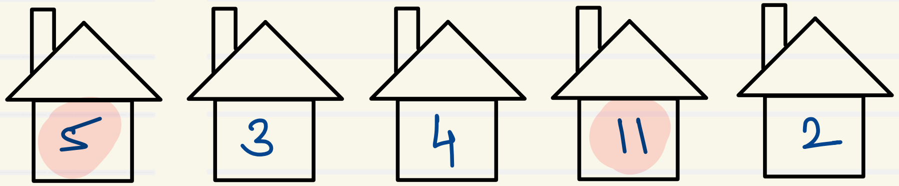
- Rob the houses to get maximum loot but no adjacent houses can be robbed.



$1+3=4$

$2+1=3$

$1+1=2$



$5+4+2=11$

$5+11=16$

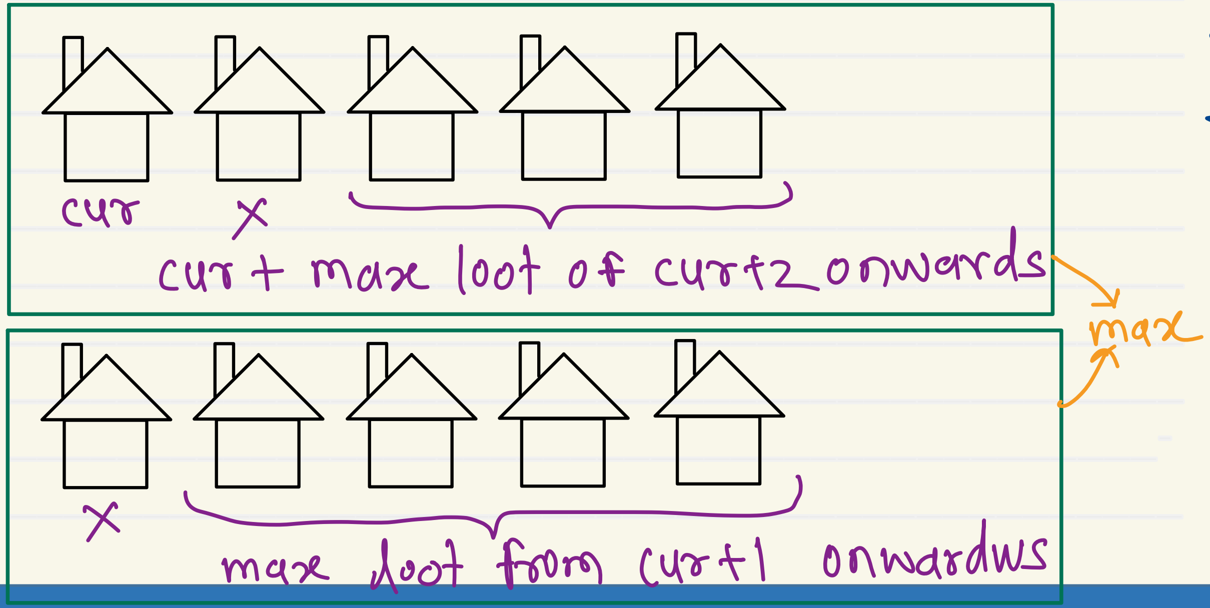
$5+2=7$

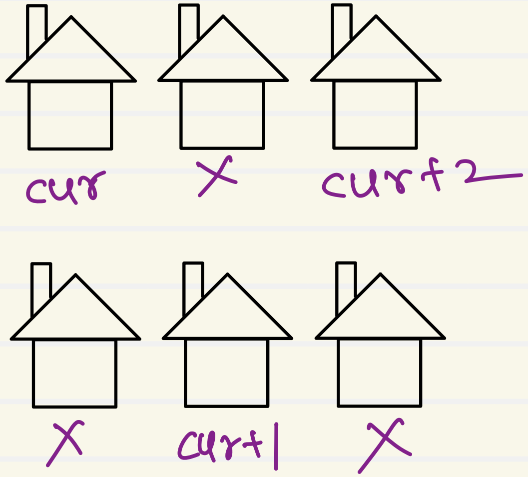
$3+11=14$

$3+2=5$

$4+2=6$

$4+5=9$





- Rob the houses to get maximum loot but no adjacent houses can be robbed.

$$loot = max(cur + maxLoot[cur+2], maxLoot[cur+1])$$

0	1	2	3	4	5	6
16	14	11	11	2	0	0

$$loot = max(2+0, 0) = 2$$

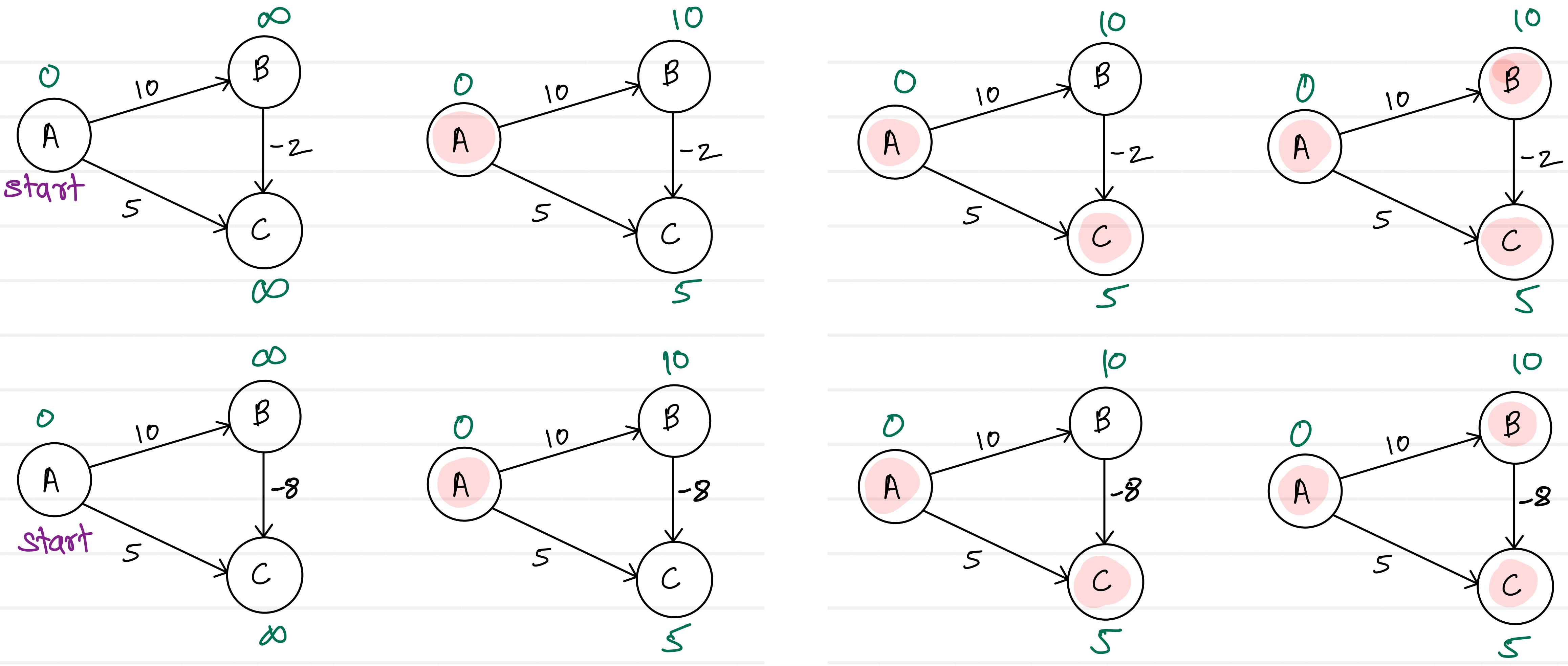
$$loot = max(11+0, 2) = 11$$

$$loot = max(4+2, 11) = 11$$

$$loot = max(3+11, 11) = 14$$

$$loot = max(5+11, 14) = 16$$

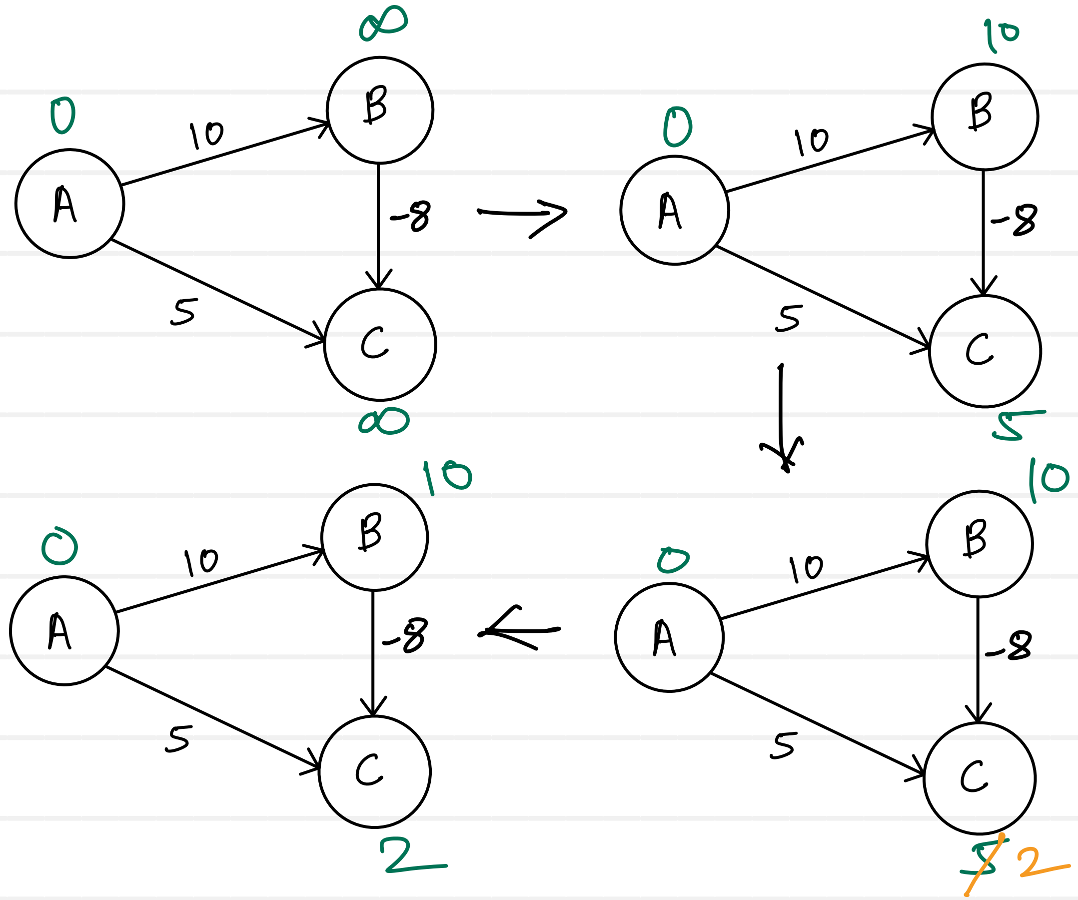
Dijkstra's Algorithm



if graph is having -ve edges, Dijkstra's algo may fail

Bellman Ford Algorithm

1. Initializes distances from the source to all vertices as infinite and distance to the source itself as 0.
2. Calculates shortest distance V-1 times: (V-1)
   For each edge u-v, → (E)
   if dist[v] > dist[u] + weight of edge u-v,
   then update dist[v], so that
   dist[v] = dist[u] + weight of edge u-v.
3. Check if negative edge cycle in the graph:
   For each edge u-v, → (E)
   if dist[v] > dist[u] + weight of edge (u,v),
   then graph has -ve weight cycle.



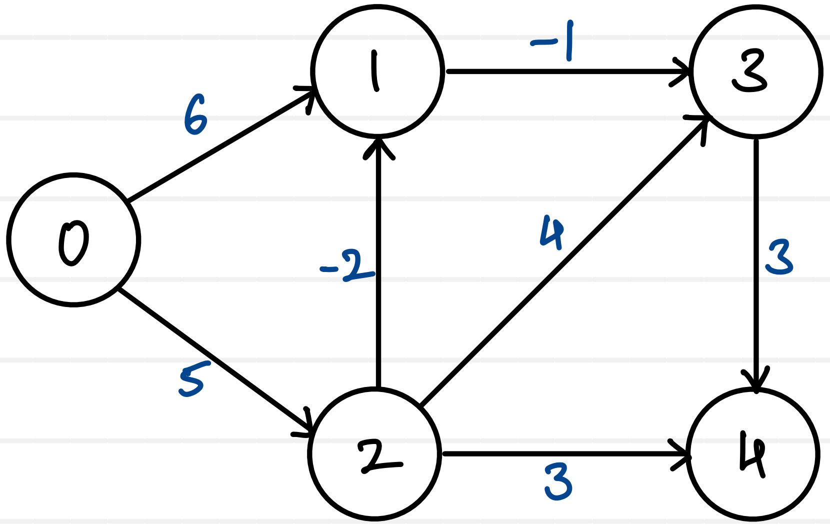
Time ∝ (V-1)E + E

Time ∝ VE - ~~E~~ + ~~E~~

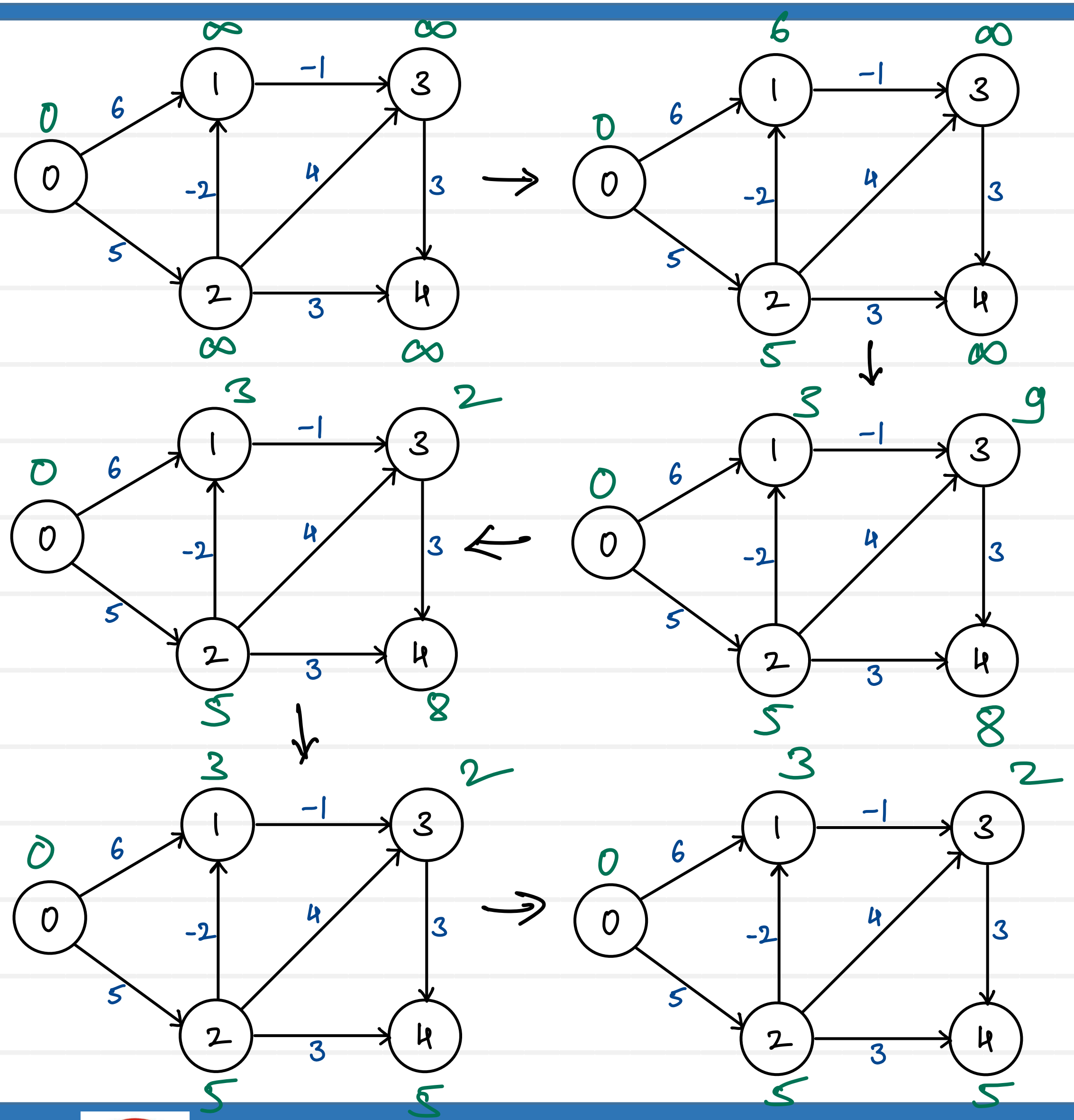
$\boxed{T(V,E) = O(VE)}$

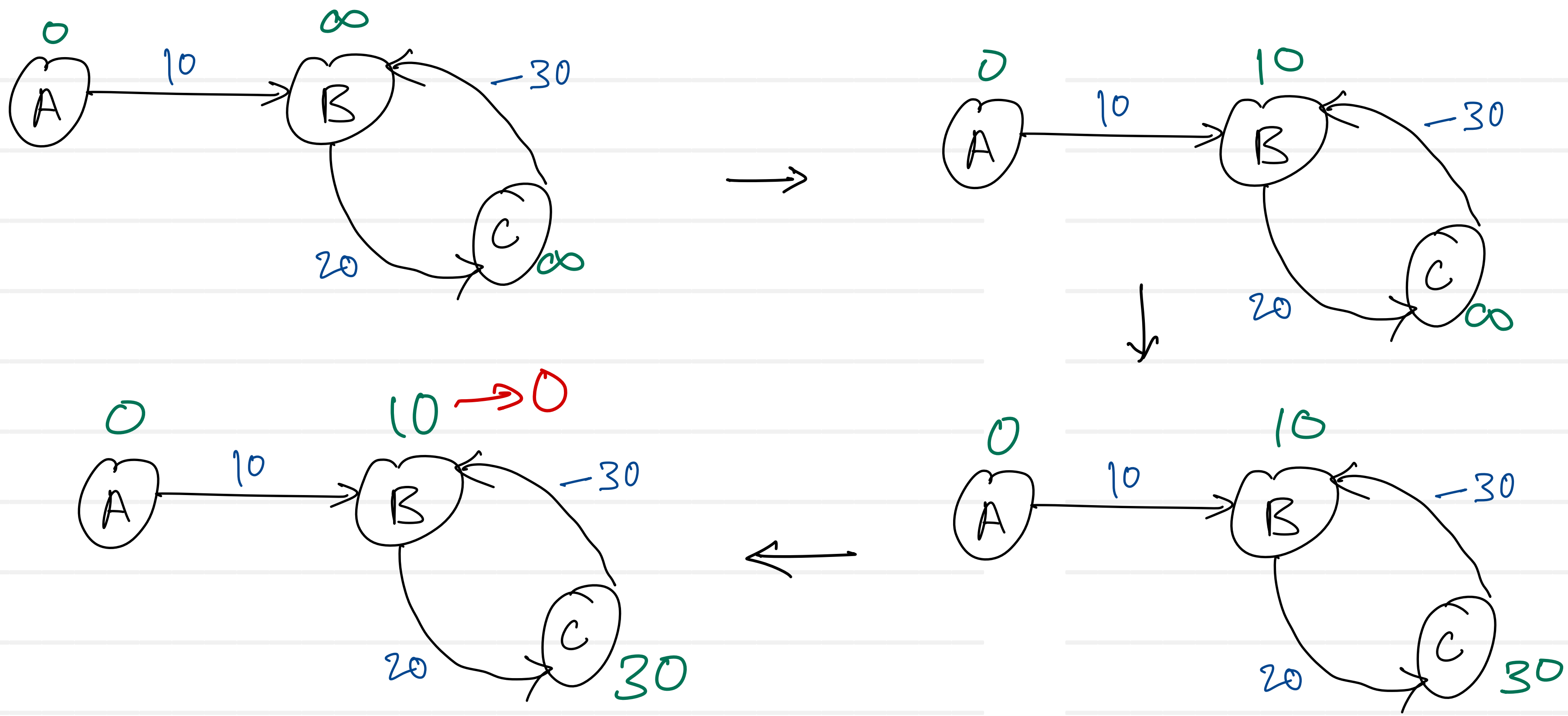
$\boxed{S(V) = O(V)}$

Bellman Ford Algorithm



	0	1	2	3	4
	0	∞	∞	∞	∞
Pass 1	0	6	5	∞	∞
Pass 2	0	3	5	9	8
Pass 3	0	3	5	2	8
Pass 4	0	3	5	2	5





cycle wt = 20 + -30

= -10

is -ve edge cycle