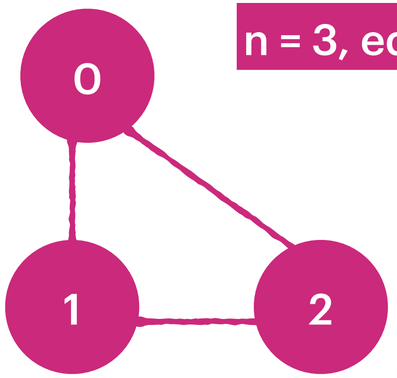
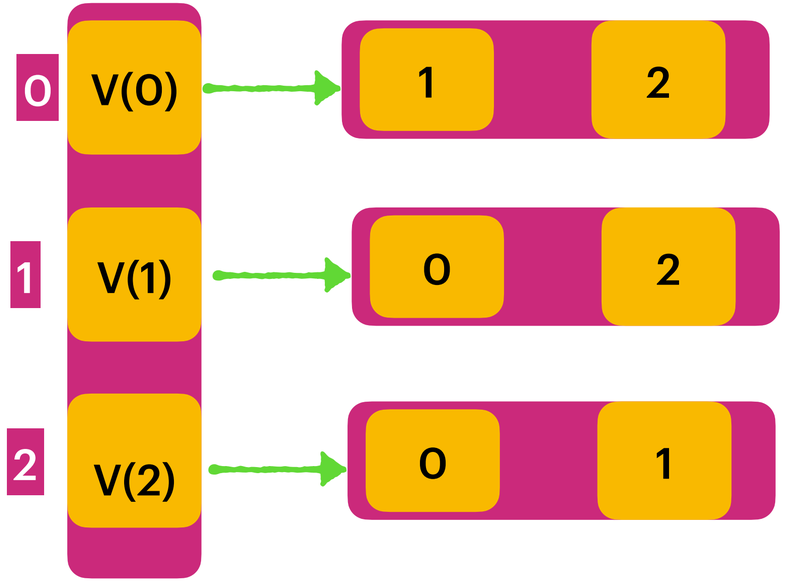


n = 3, edges = [[0,1],[0,2],[1,2],[2,1],[2,0],[1,0]]

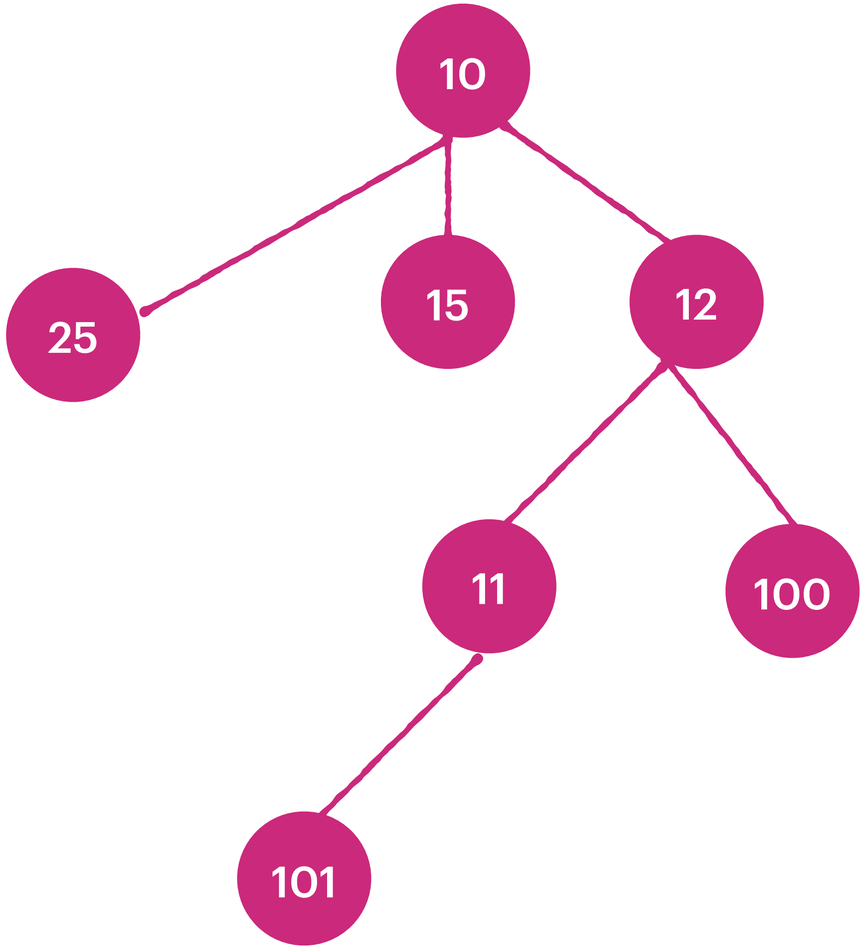
List< LinkedHashSet >



Is this can be applied in RealWorld ?
Limitations ***
1. Graph should be fixed .
2. Demands vertexes must Be in the range of 0 to n-1
3. Occupy the fixed memory.

Hashing

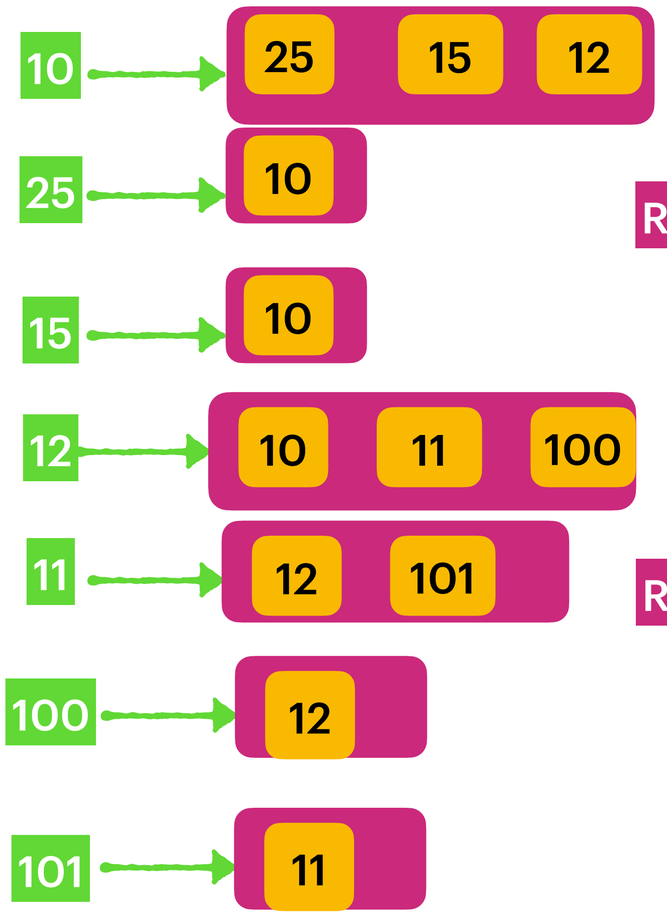
Add Vertex	O(1)
Remove Vertex	O(V)
Add Edge	O(1)
Remove Edge	O(1)
Print	O(V+E)




```
10 => [25, 15, 12]
25 => [10]
15 => [10]
12 => [10, 11, 100]
11 => [12, 101]
100 => [12]
101 => [11]
```

Map<VertexNumber, Connections>
Map<Integer, Set<Integer>>

Hashing



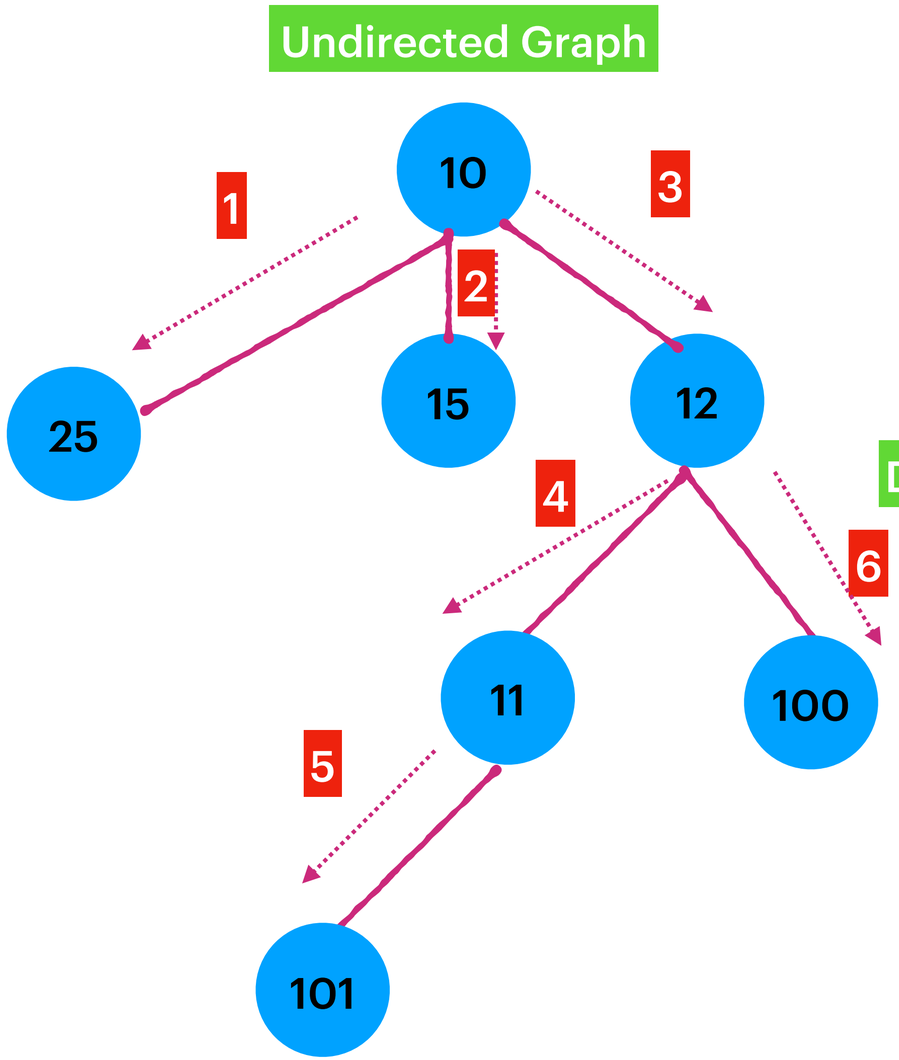
Add Vertex O(1)

Remove Vertex O(E)

Add Edge O(1)

Remove Edge O(1)

Print O(V+E)



Depth First Search

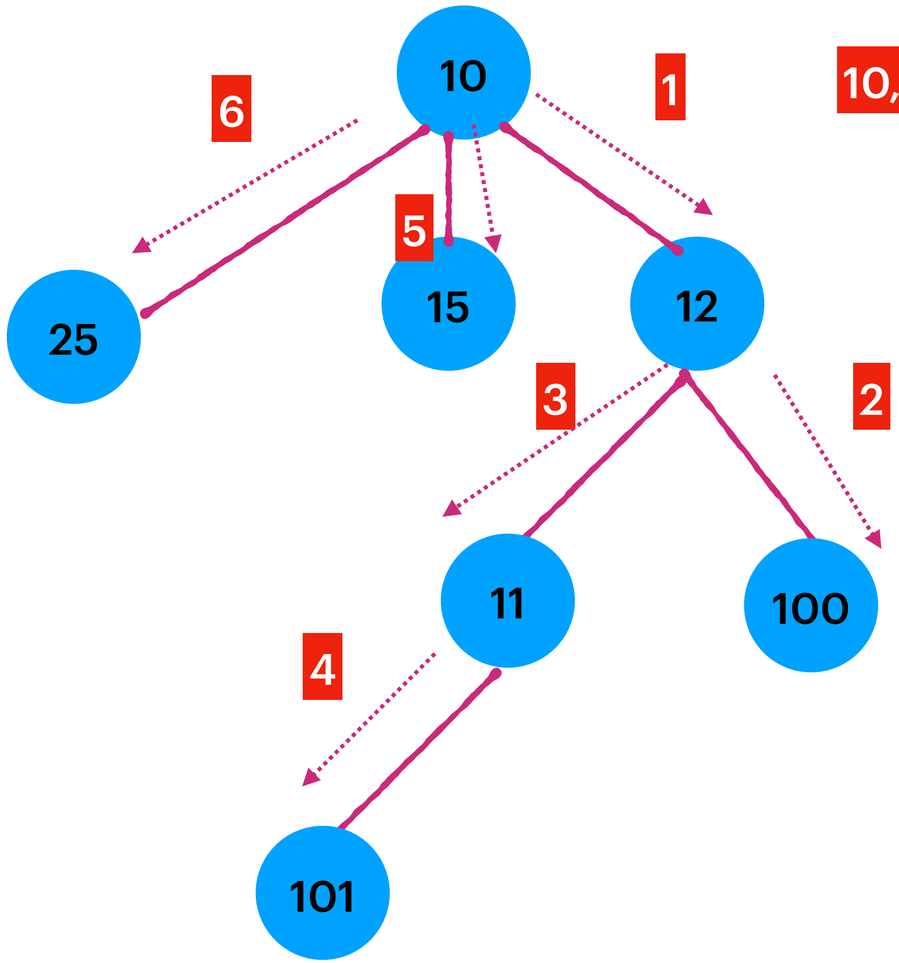
Moving => from root- left to right

10,25,15,12,11,101,100

Undirected Graph

Depth First Search

Moving => from root - right to left



10, 12,100,11,101,15,25