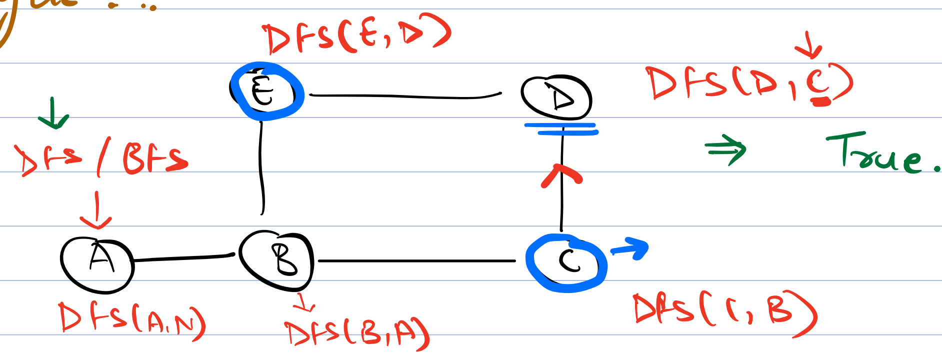
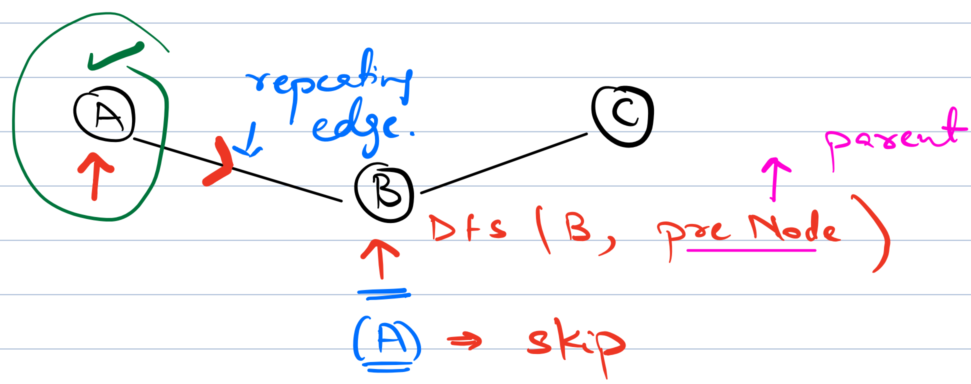
Q. Given an undirected graph.

Return True if the graph contains a cycle . . .

Eg





Code

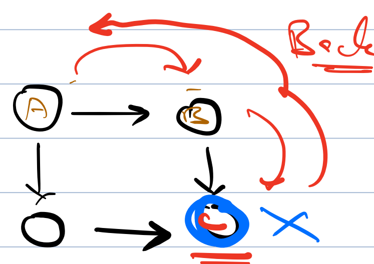
```
bool isCyclic ( u , parentNode) {
        visited [u] = true;
        for ( all v connected to u) {
            if ( visited [v] == True &&
                 v != parentNode) {
                    return True;
```

(isCyclic → DFS; u → Source; parentNode → NULL)

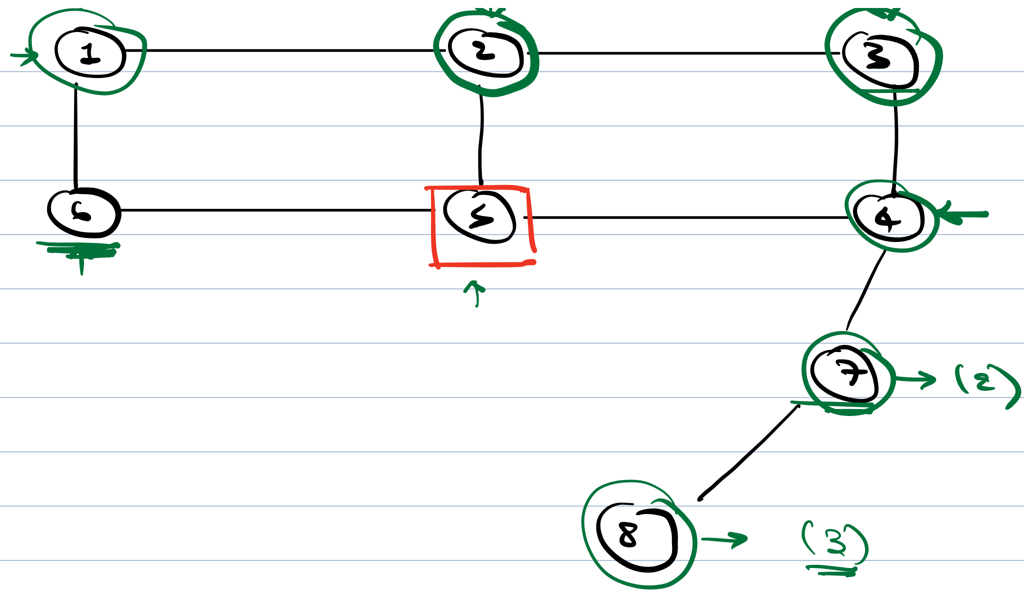
```
            }
            else if ( visited [v] == false) {
                if ( isCyclic (v, u)) {
                }          return True;
            }
        }

        return false;

}
```

T.C. = O(V+E)

H.W. Directed graph.



Q Given an undirected / unweighted graph.
Find min distance from a source to all nodes.

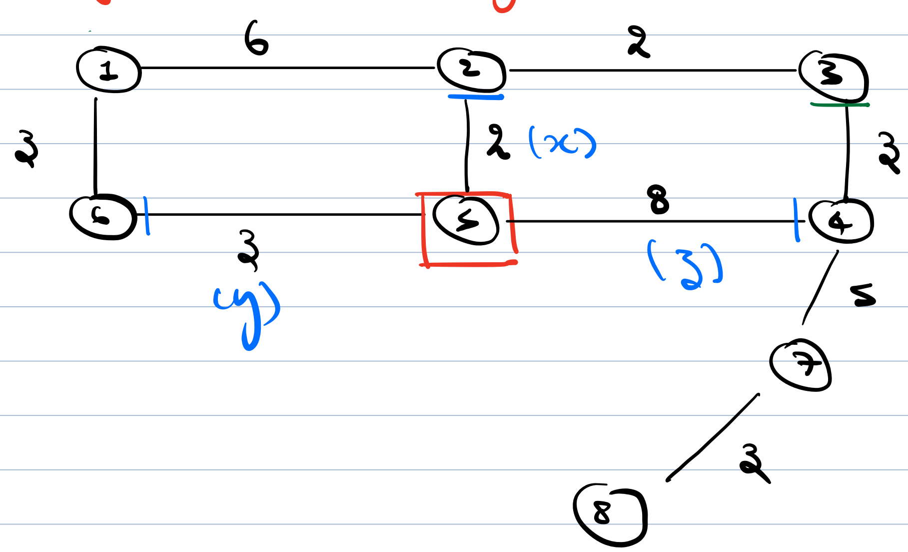


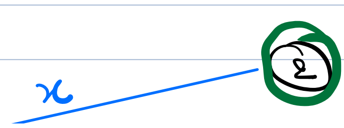
O/P

0	1	2	3	4	5	6	7	8
	2	1	2	1	0	1	2	3

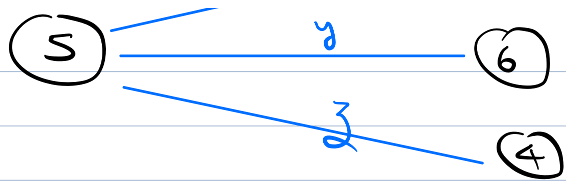
Undirected / Weighted. (No -ve weights)

(Dijikstra's Algo)



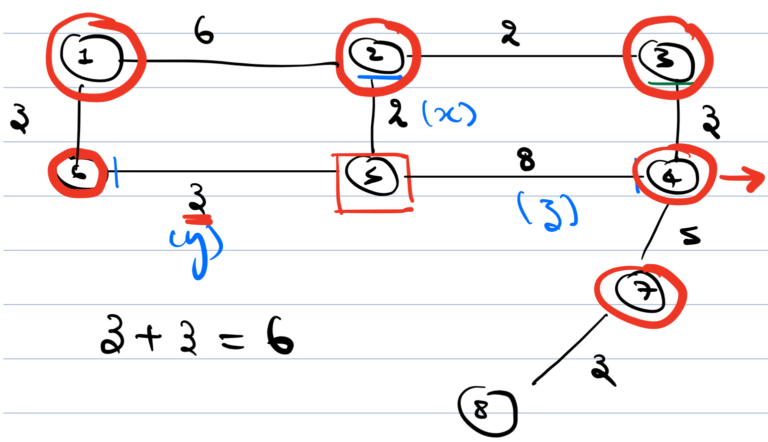


$x = \min(x, y, z)$



Ways to reach 2 is
1) Direct (x)
2) Through 6 (y + _)
3) Through 4 (z + _)

$$x = \min(x,\ y + \underline{+ve},\ z + \underline{+ve})$$



$3 + 3 = 6$

1 :	~~∞~~ ~~8~~ 6
2 :	~~∞~~ 2
3 :	~~∞~~ 4
4 :	~~∞~~ ~~8~~ 7
5 :	0
6 :	~~∞~~ 3
7 :	~~∞~~ 12
8 :	~~∞~~ 15

Min Heap ⇒ Pair < Length, Node >

L N

~~< 2, 2 >~~

~~< 3, 6 >~~

~~< 8, 4 >~~

~~< 4, 3 >~~

~~< 8, 1 >~~

~~< 6, 1 >~~

⇒ Edges

O(E) → space

~~<7, 4>~~

~~<12, 7>~~

<15, 8>

T.C. = $E \log(E)$

Code

Min Heap → h

MinDist [N+1] = {∞}

Min Dist [source] = 0;

void dijikstra (source) {

for (all v connected to source) {

h.add(Pair< W_{s-v} , v>);

MinDistance [v] = W_{s-v};

}

while (! h.isEmpty()) {

Pair < l, u> = h.getMin();

if (l < MinDistance [u]) {

for (all x connected to u) {

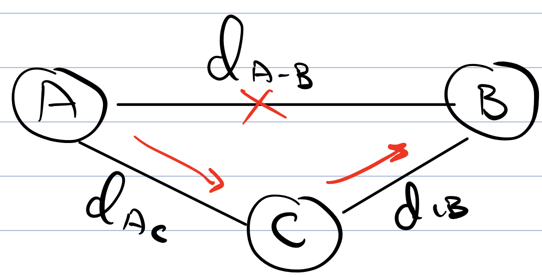
$D_{s-x} = \underset{D_{s-u}}{\underset{\downarrow}{l}} + W_{u-x}$;

if (D_{s-x} < MinDist [x]) {

```
MinDist[x] = Ds-x
h.insert(Pair<Ds-x, x>);
            }
          }
        }
      }
    }
```

$$T.C. = O(V + \cancel{E} + E\log E)$$

$$= O(V + E\log E)$$



$\Rightarrow$ Relaxing an edge

$$d_{AC} + d_{CB} < d_{A-B}$$

✩ Floyd Warshall Algo.

(All possible shortest path algo)

Q. Find min distance to reach any node from every node.

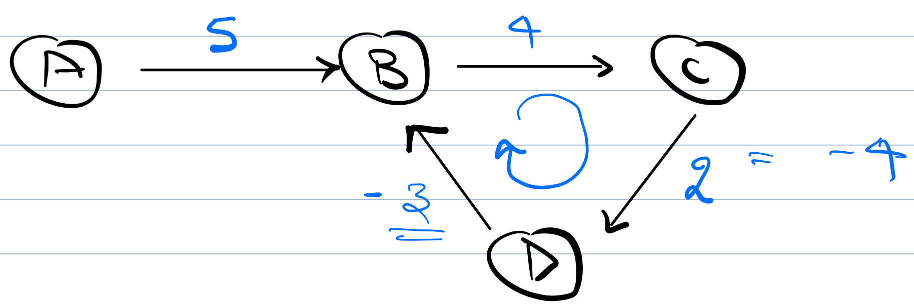
O/P → Matrix of size $V \times V$

Undirected → -ve weight edge. ✗

$A \overset{2}{\text{———}} B \overset{-4}{\text{———}} C$

$2 - 4 - 4 = -6 \quad -4 - 4 = -14$

Directed (No -ve weight cycle)



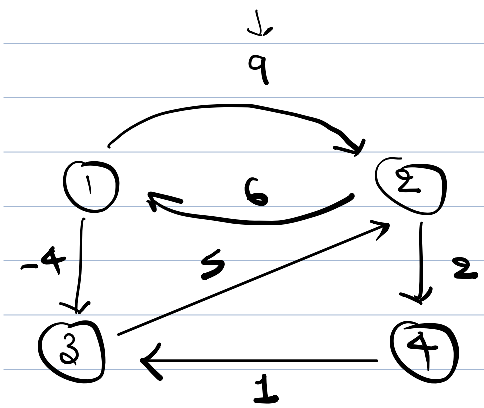
O/P ⇒ D $(V+1) \times (V+1)$ matrix

$D[i][j]$ → Min distance from i to j

Iterate over all nodes

↳ Treat this node as int. node

↳ Try to relax every edge.

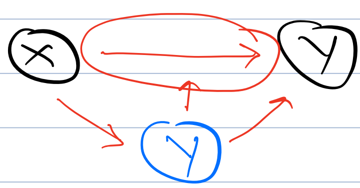


Adjacency matrix

	1	2	3	4
1	0	9	-4	∞
2	6	0	∞	2
3	∞	5	0	∞
4	∞	∞	1	0

D_0

1) Node 1 as inter node.



2, 4 → 2, 1 + 1, 4

(2) (6) ∞

	1	2	3	4
1	0	9	-4	∞
2	6	0	∞	2
3	∞	5	0	∞
4	∞	∞	1	0

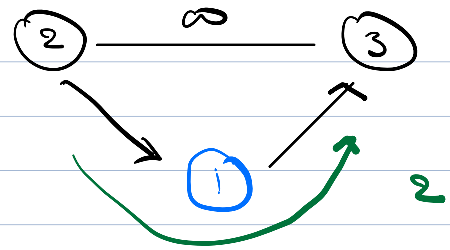
D_0

→

	1	2	3	4
1	0	9	-4	∞
2	6	0	2	2
3	∞	5	0	∞
4	∞	∞	1	0

D_1

$M[2][3] \quad , \quad \overset{6}{M[2][1]} + \overset{-4}{M[1][3]} = 2$



$M[3][2] \quad , \quad (M[3][1] + M[1][2])$

$\le \qquad\qquad \infty$

$M[3][4] \quad , \quad (M[3][1] + M[1][4])$

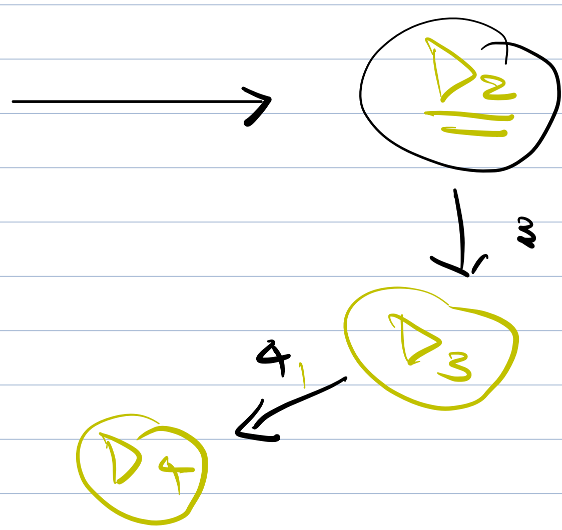
$\infty \qquad\qquad \infty$

$M[4][2] \quad , \quad (M[4][1] + M[1][2])$

2) Node 2 as Int Node

	1	2	3	4
1	0	9	−4	∞
2	6	0	2	2
3	∞	5	0	∞
4	∞	∞	1	0

D_1



Code

T.C. = $O(V^3)$

Graph 4 → 1) Topological Sort
2) Bipartite graph (Graph colour)

5 → Prim's / Kruskal's (MST)