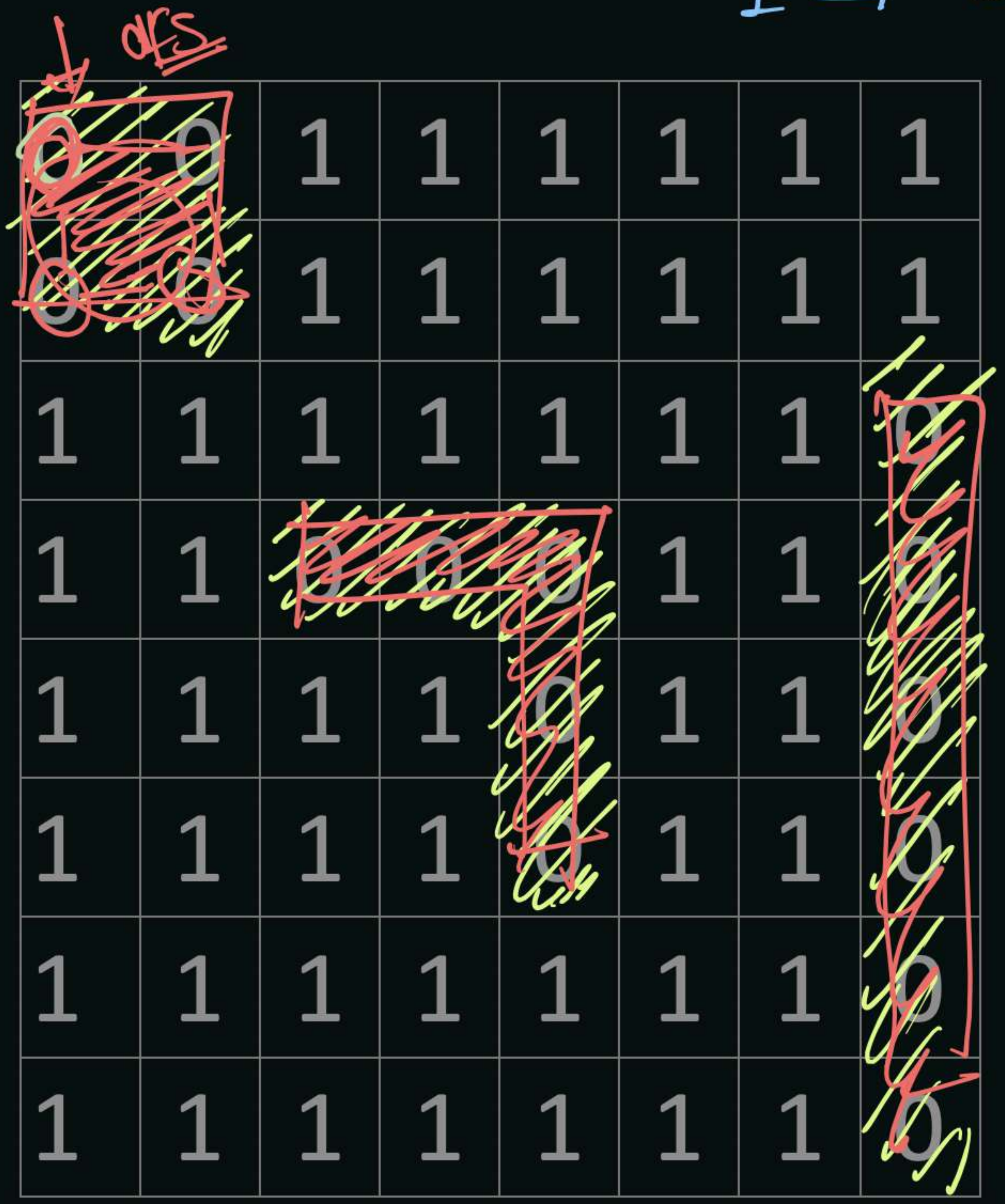
Number of Islands

Thursday, 8 July 2021 6:59 PM

$0 \to$ lands

$1 \to$ water

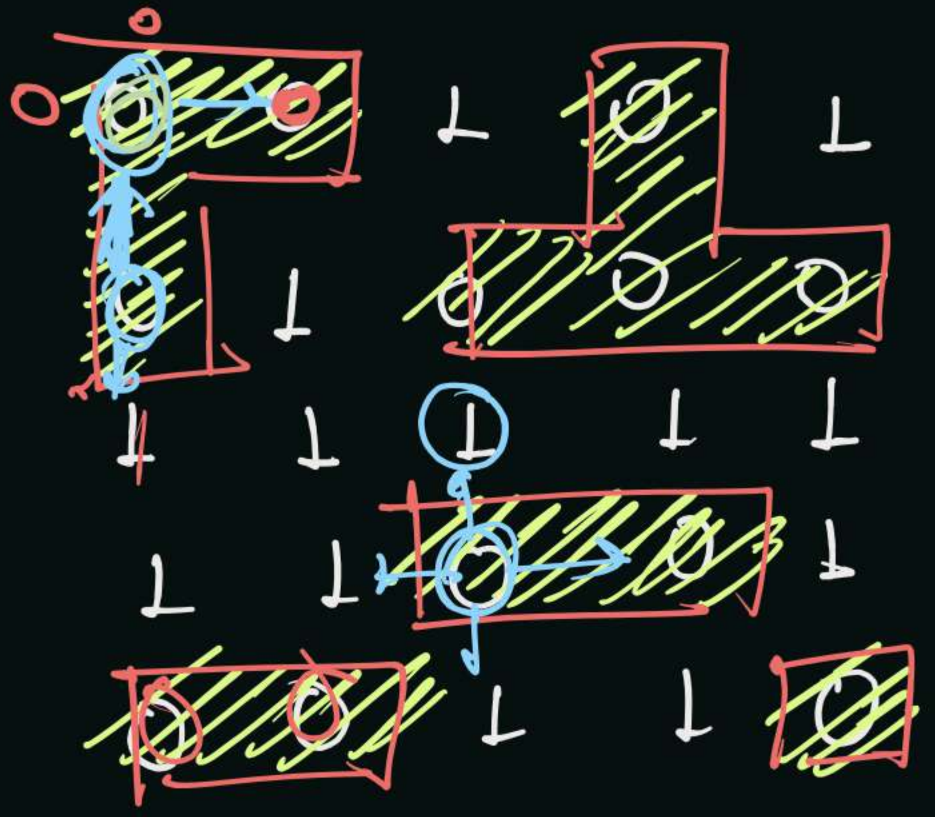
find no. of islands.



0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	0
1	1	0	0	0	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0

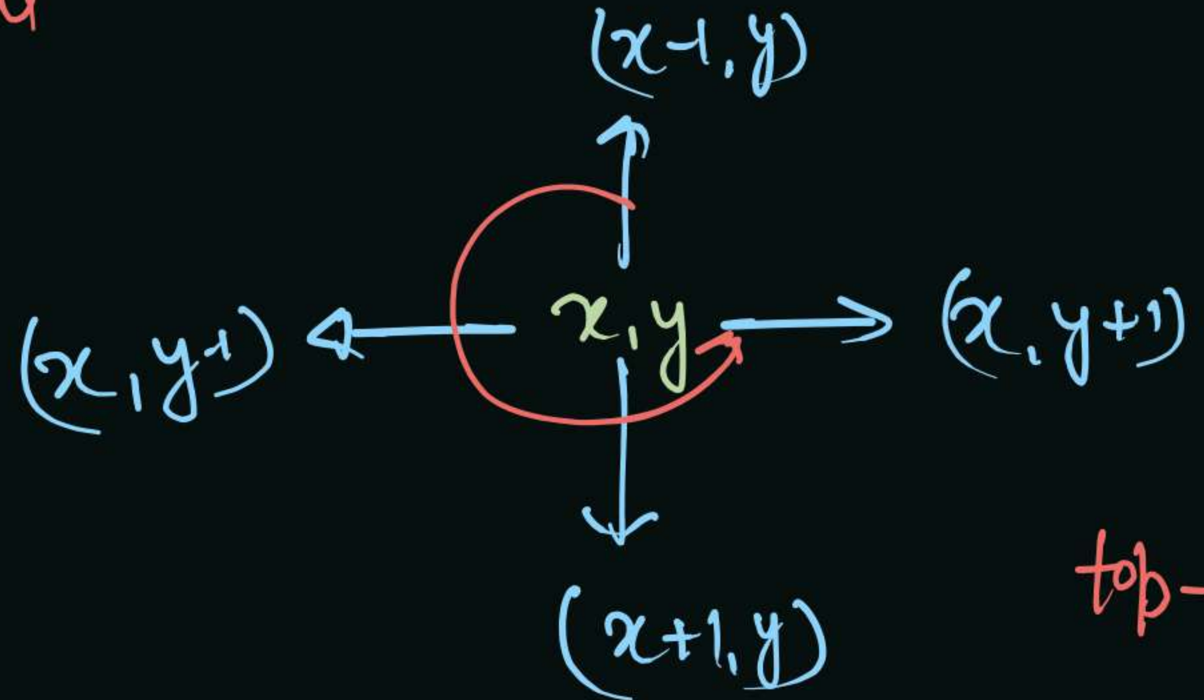
no. of islands = 3

NOTE: Lands are horizontally and vertically connected.



0	0	1	0	1
0	1	0	0	0
1	1	1	1	1
1	1	0	0	1
0	0	1	1	0

no. of islands = 5



for 4 Edges ⟶

conditions for edges

① → valid index

② → land,

i.e. graph[x][y] == 0

top → left → down → right

✓ marking - ~~unmarking~~ ✗

↪ graph[x][y] = -1

Perfect Friends

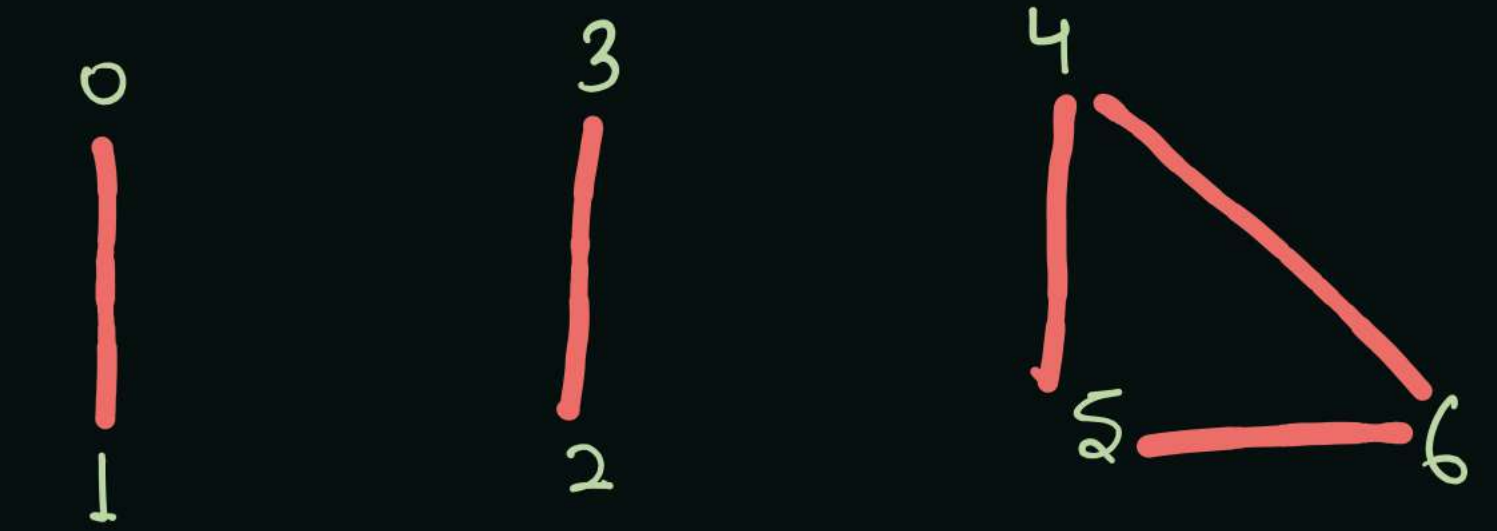
Thursday, 8 July 2021 6:59 PM

7 → no. of vertices → no. of students
5 → no. of Edges → pairs of students

0 1 ✓
2 3 ✓
4 5 ✓
5 6 ✓
4 6 ✓

} these are Edges

clubs = 3



0 → 1
1 → 0
2 → 3
3 → 2
4 → 5 | 6
5 → 4 | 6
6 → 4 | 5

count ll pairing of students is such manner so that no two students are from same house.

{ {0, 1}, {2, 3}, {4, 5, 6} } } connected components

count in single club

a	b	c	d
4	4	1	
a×b	b×c	c×d	
a×c	b×d		
a×d			

0 − 2
0 − 3
1 → 2
1 → 3
] 4

$\boxed{4}$

0 − 4
0 − 5
0 − 6
1 → 4
1 − 5
1 − 6
]

$\boxed{6}$

2 − 4
2 − 5
2 − 6
3 − 4
3 − 5
3 − 6
]

$\boxed{6}$

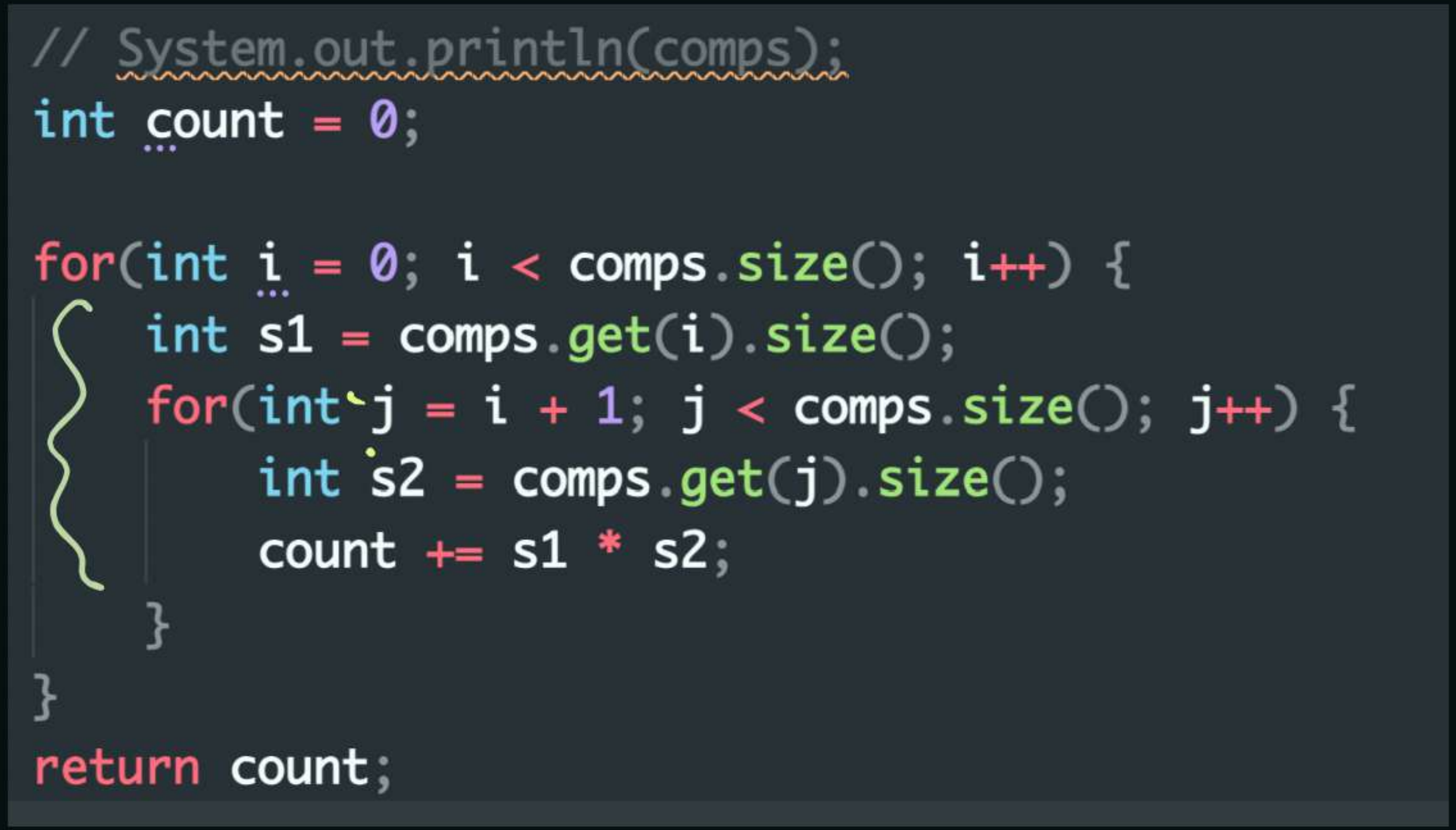
$\boxed{16}$

Arraylist<Integer>[] graph

[[0, 1], [2, 3], [4, 5, 6]]

n = no. of components

complexity to find pairs ??



```
// System.out.println(comps);
int count = 0;

for(int i = 0; i < comps.size(); i++) {
    int s1 = comps.get(i).size();
    for(int j = i + 1; j < comps.size(); j++) {
        int s2 = comps.get(j).size();
        count += s1 * s2;
    }
}
return count;
```

size of Components

a	b	c	d	e	f
a×b + a×c + a×d + a×e + a×f	+ b×c + b×d + b×e + b×f	+ c×d + c×e + c×f	+ d×e + d×f	+ e×f	$\ominus$

sum=0

steps ① res = res + size of ith comp × sum.

sum = sum + size of ith comp

direction of traversal ←

Time → O(n)

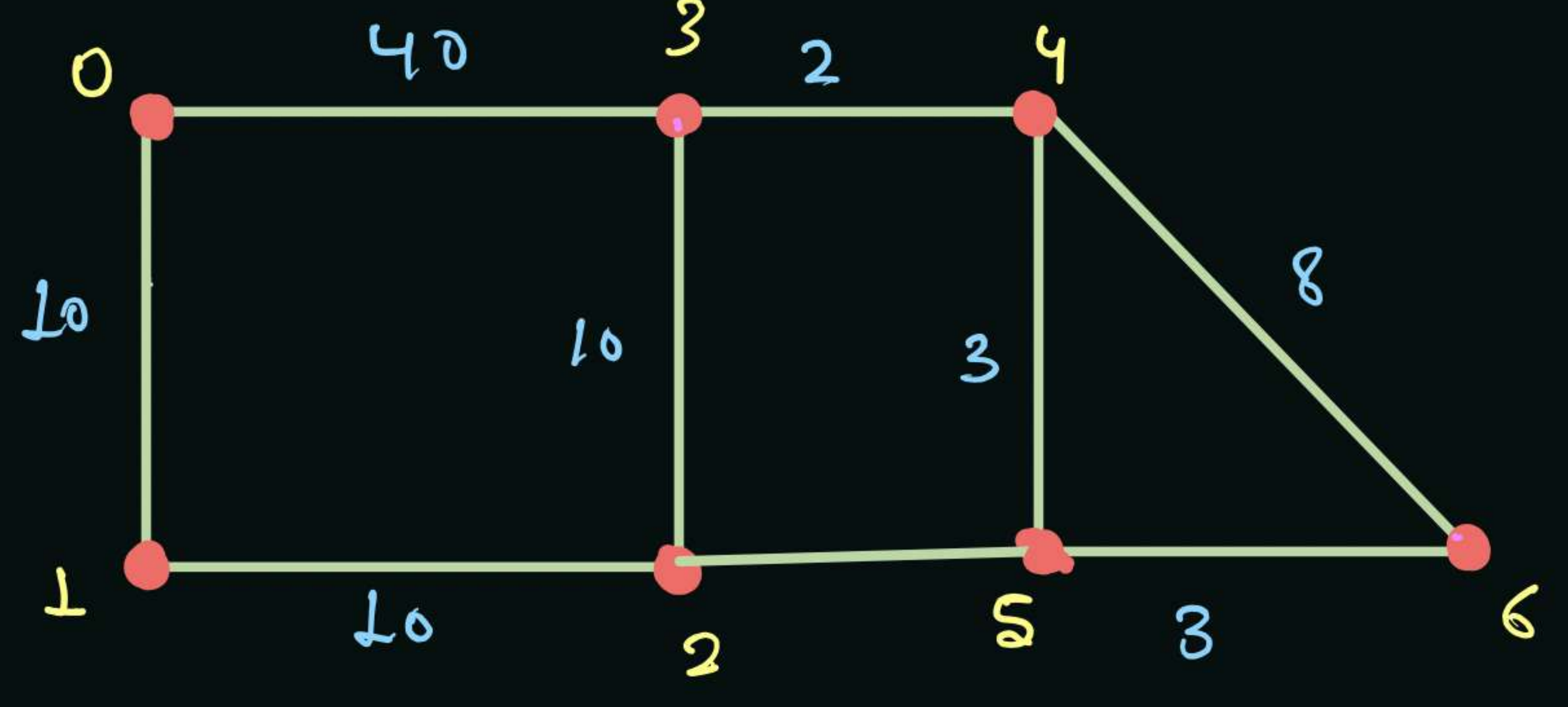
a(b+c+d+e+f) + b(c+d+e+f) + c(d+e+f) + d(e+f) + e(f) + $\ominus$

← - - - c+ c(d+e+f) d(e+f) ← e×f + 0×0

Hamiltonian Paths and Cycles

Thursday, 8 July 2021 6:59 PM

Source dependent problem



Hamiltonian Path :- path from which we can visit all the vertices without visiting any vertex twice.

Hamiltonian cycle :- Hamiltonian path in which we have a back edge from last visited src to original source.

Hamiltonian path →

✓ 5 → 6 → 4 → 3 → 2 → 1 → 0

Hamiltonian cycle →

5 → 6 → 4 → 3 → 0 → 1 → 2 *

5 → 2 → 1 → 0 → 3 → 4 → 6 *

Edge

Edge

cycle → path *

path ← path.

Hint → keep original src., why??

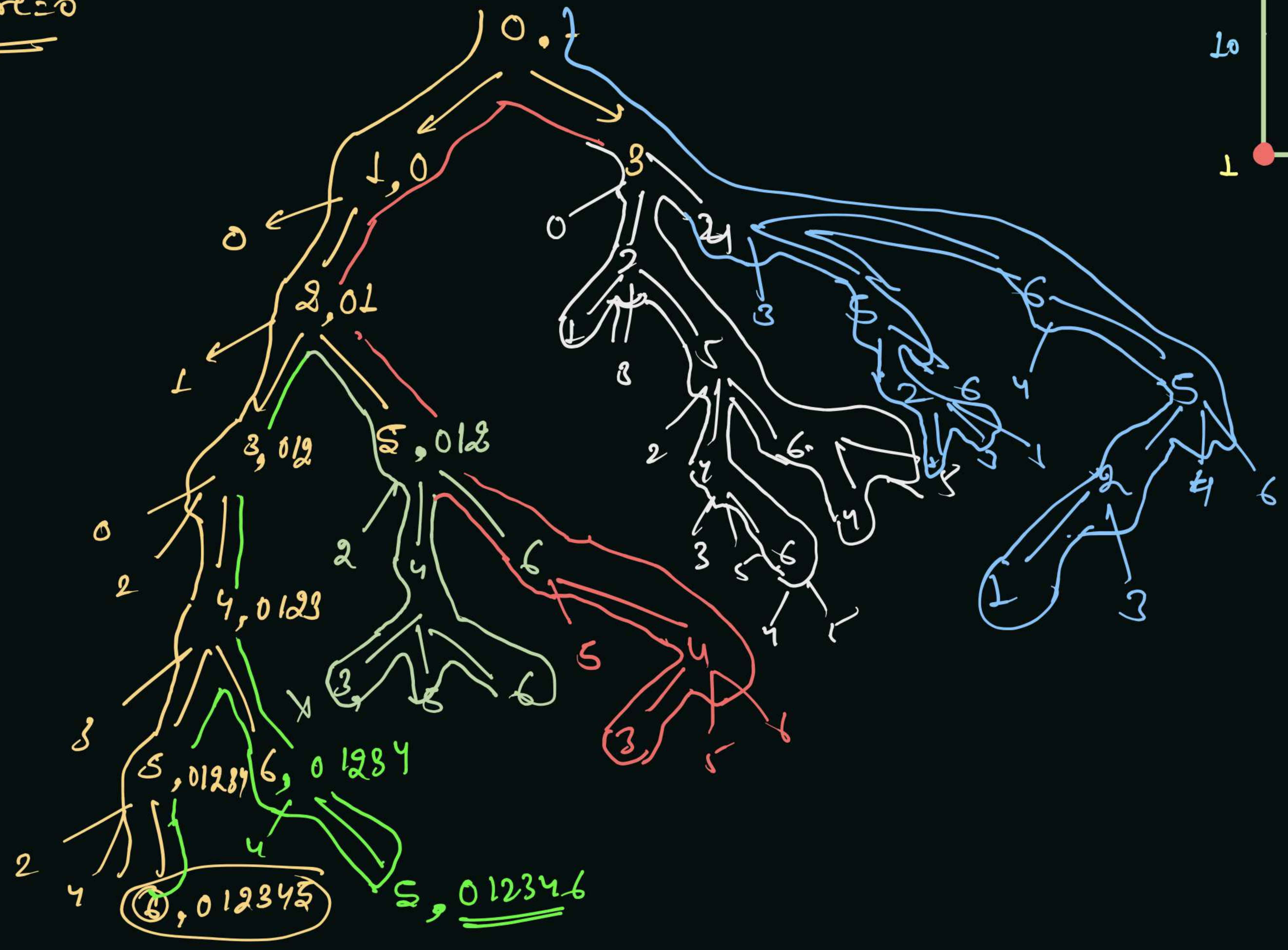
src=0

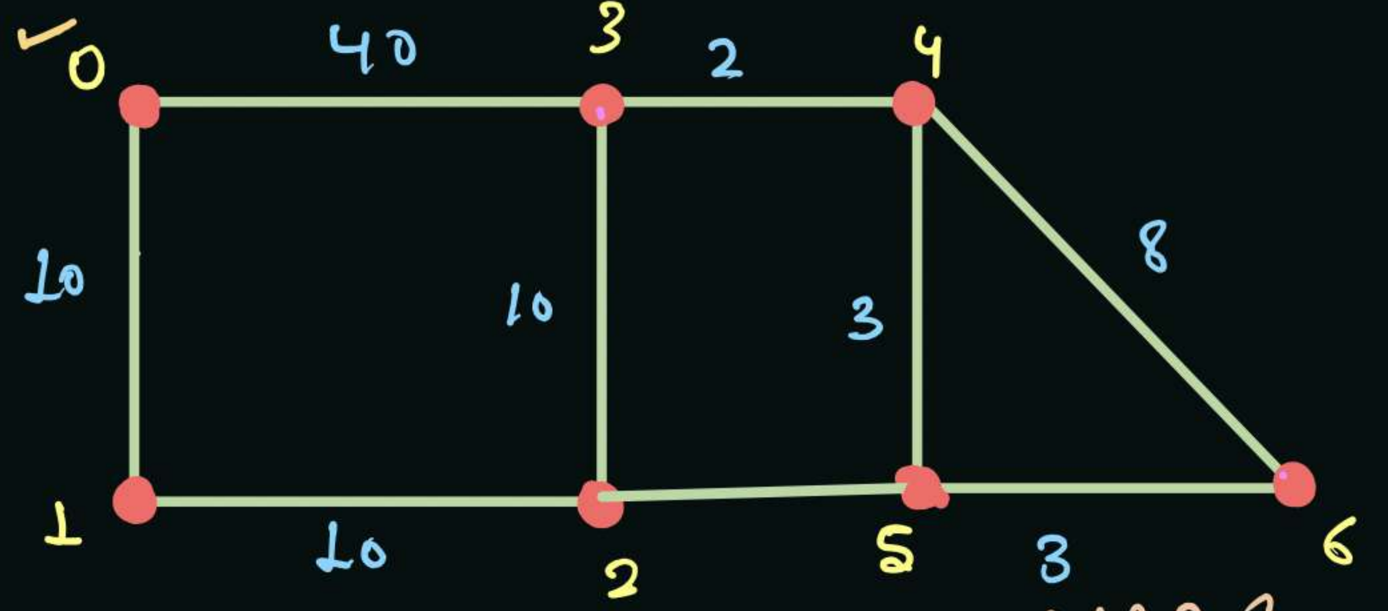
osrc=0

src, psf, osrc

mark ✓

unmark ✓





nbr == src } BackEdge

psf. length = n-1

✓ 0 1 2 3 4 5 6

✓ 0 1 2 3 4 6 5

✓ 0 1 2 5 6 4 3

✓ 0 3 4 6 5 2 1

To check cycle

Breadth First Search

Thursday, 8 July 2021 6:59 PM

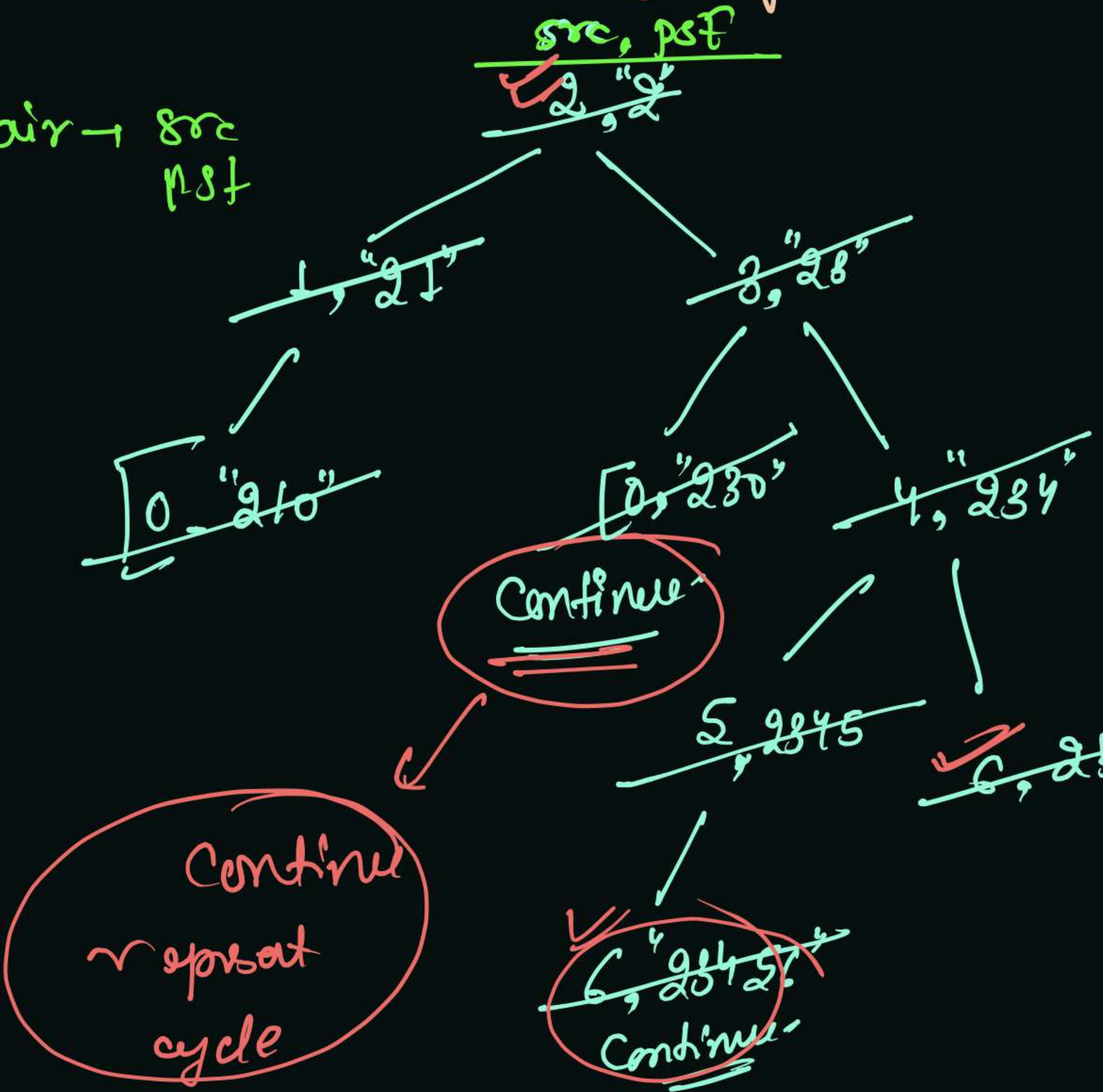
Src- is Given -(In terms in Edge)
Smallest Dist

src = 2.

Smallest path of all the vertex from given source in terms of Edge.

Data Structure Required – Queue

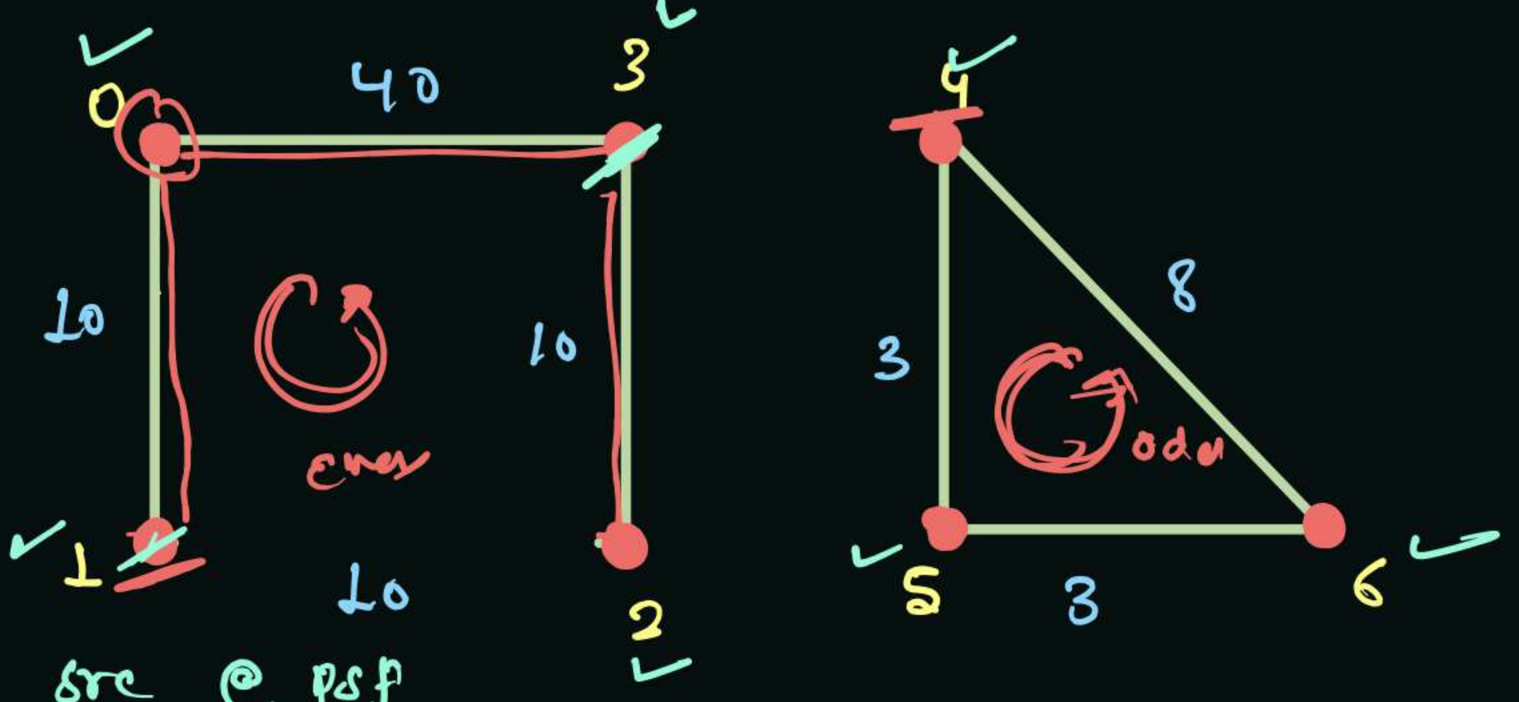
B Pair → src
psf



marking →

unmark → mark

marked → continue.



src	@	psf
2	@	2
1	@	21
3	@	23
0	@	210
4	@	234
5	@	2345
6	@	2346

Steps.

1. Get + Remove
2. mark*] Level order
3. work → print
4. Add Neighbour unvisited

Searching Algo in Graph →

Depth First Search	Breadth First Searching
① Data structure → Stack	① Data Structure - Queue
② Search in Depth first	② Search level wise, or, radially search
③ Helpful in gcc concept and all paths concept.	③ Helpful in level wise traversal
	④ Smallest distance b/w src to dst in terms of Edges.
④ → topological sort	⑤ → Dijkstra's, etc.

Is Graph Cyclic

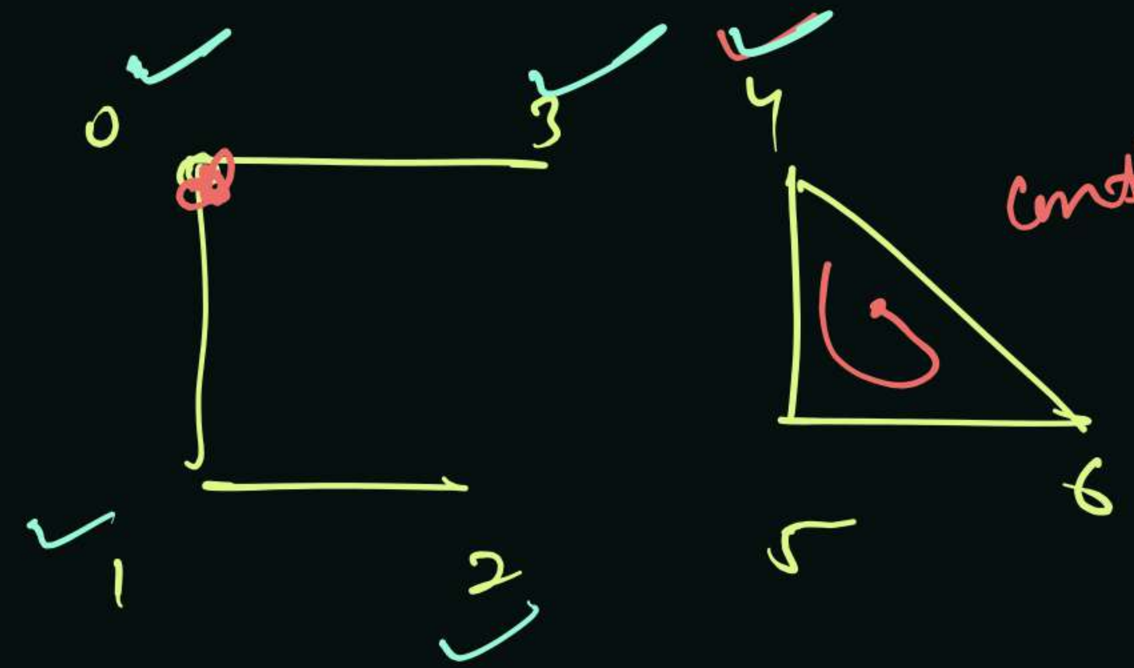
Thursday, 8 July 2021 7:00 PM

for graph

cyclic

1. main function → iterate on vtx and call to BFS (visited)

2. Helper function → BFS (src recieved from main cyclic function)



continue → return true