Date   Page

Day - 212

Graph - 16

* **Shortest source to destination path:**

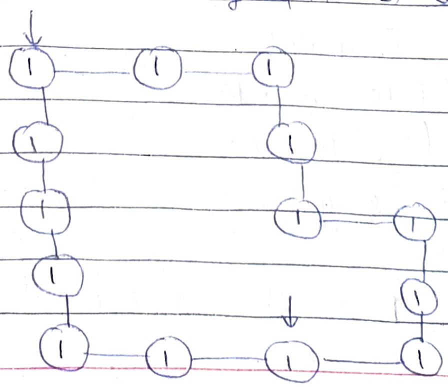
	0	1	2	3
0	$\textcircled{1}$	1	1	0
1	1	0	1	0
2	1	0	1	1
3	1	0	0	1
4	1	1	$\textcircled{1}$	1

⇒ So, we have to find the shortest distance between ~~target~~ source & destination.

⇒ But we can travel only in left, right, up & down '1' to reach the destination.

⇒ So, how to think that this is a graph problem.

⇒ First, make the graph of above problem -



⇒ So, we know how to find shortest path ~~b/w~~ in a undirected unweighted graph.

⇒ Simply, we can use BFS.

⇒ BFS works on traversing their children.

⇒ We can also apply dijkstra but it increases the time complexity.

BFS → $O(V+E)$

Dijkstra → $O(E \log V)$

⇒ We will take a visited matrix & start from source as we do in bfs.

⇒ So, we will take a queue that contain row no., col no. & step.

<u>Time Complexity:</u> $O(n*m)$

<u>Space Complexity:</u>

⇒ $O(\min(N, M))$

Date ________ Page ________

* **Knight Walk:**

=)

	0	1	2	3	4	5	6	7
0		2	1	2			5	
1	1		3		♘		4	
2		1	2	1		3		4
3	2			3	2	4		
4		2		2		3		4
5			3		3		4	
6								
7		Tar						

⇒ Knight walk 2.5 steps —
2 - straight steps
1 - After that left or right.

=) ○ → 1 step
◯ → 2 step
◎ → 3 step

=) So, we have reached the target in 3 steps.

=) So, we will use BFS.

=) If we want to find min step & every step has constant value then we have to use BFS.

=) Row & column values —

```
int row[8] = {2, 2, -2, 2, 1, -1, 1, -1}
int col[8] = {1, -1, 1, -1, 2, 2, -2, -2}
```