

Graphs - I



Introduction to Graphs Data Structure



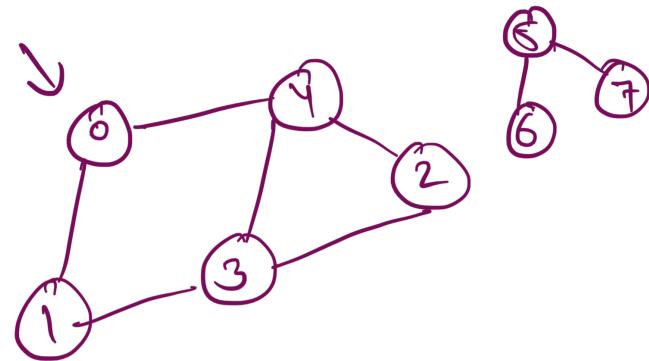
Implementation of Graphs Data Structure

DFS Graph Traversal

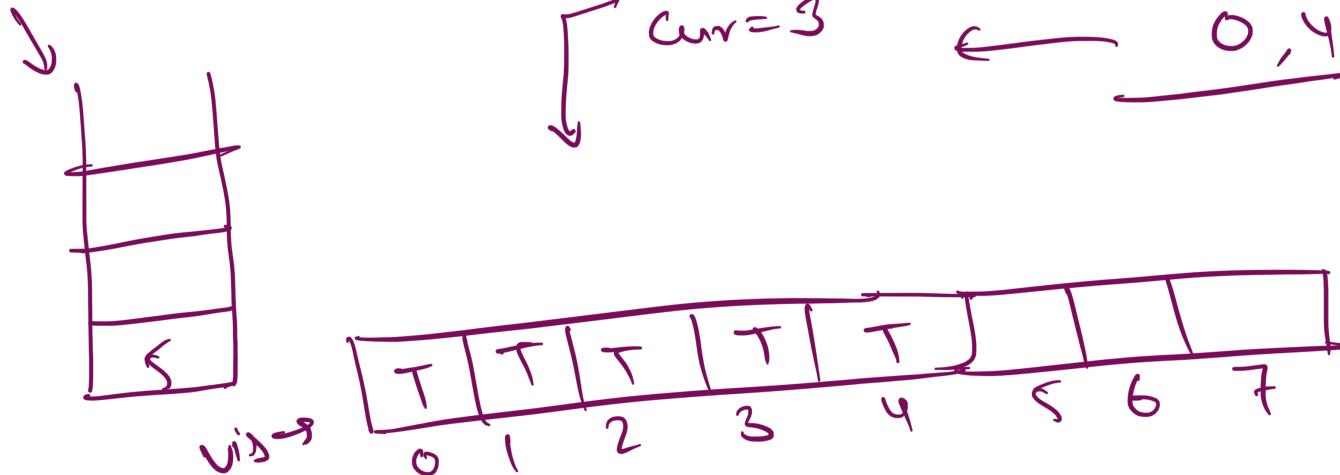
Using Recursion ✓

Using stack.

curr = 3



0, 4, 2, 3, -1





BFS Graph Traversal



Queue.



Practice Problems

1. Learn More about Graphs from: <https://www.programiz.com/dsa/graph>

Graphs - II

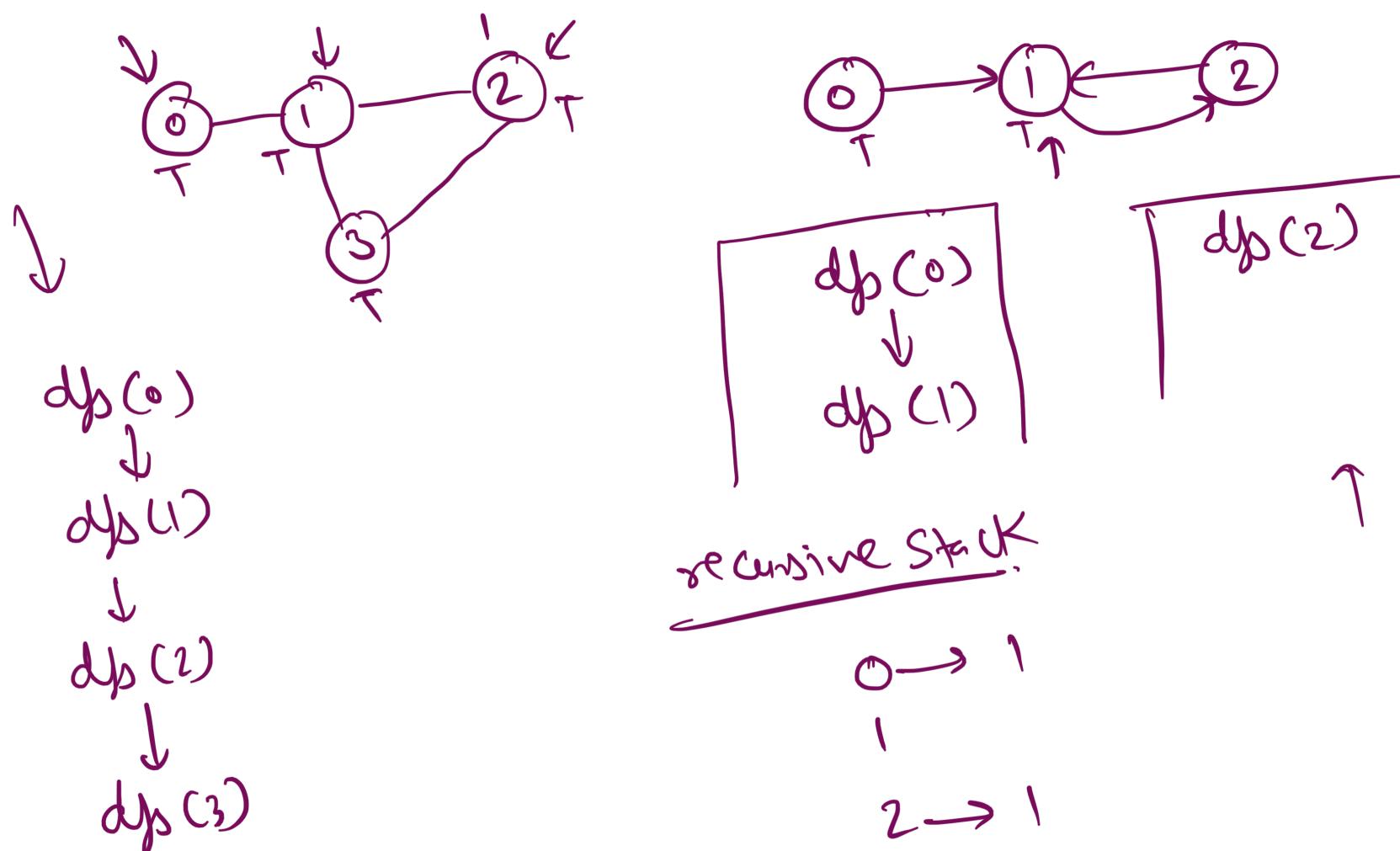


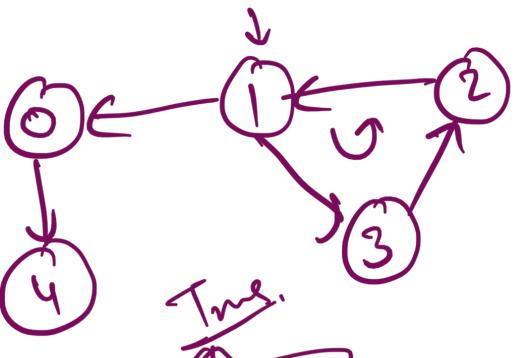
Number of connected components in a Graph



Shortest Path in a Graph

UnDetect cycle in an Directed Graph





$dps(0)$

\downarrow

$dps(4)$

$dps(1) \leftarrow$

\downarrow

True

$dps(3) \leftarrow$

\downarrow

True

$dps(2) \leftarrow$

\downarrow

True

dps

$[$

$T | T | T | T | T | T$

$0 | 1 | 2 | 3 | 4$

$] \rightarrow$

rsc

$[$

$F | T | T | T | F$

$0 | 1 | 2 | 3 | 4$

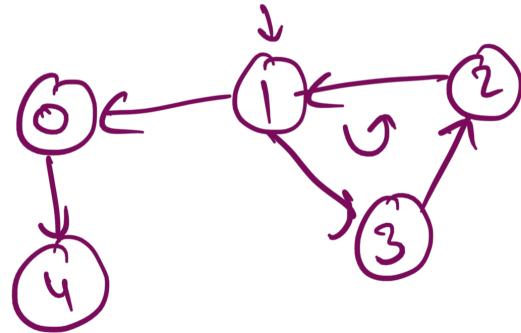
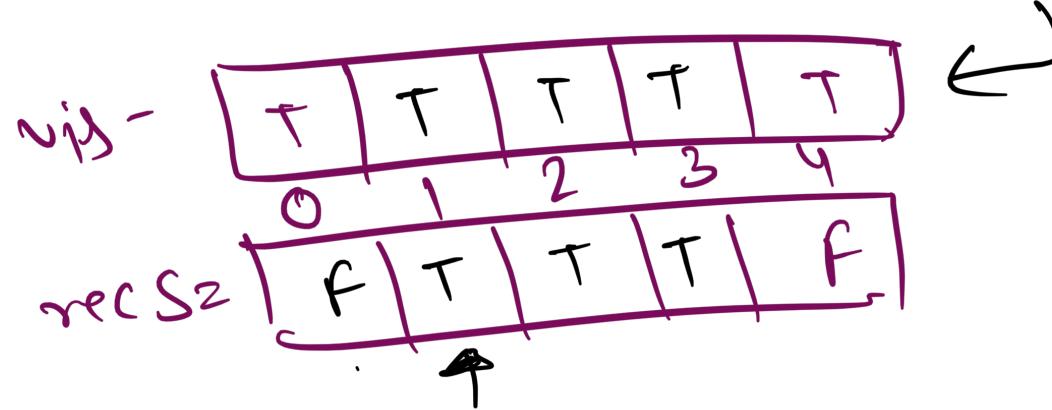
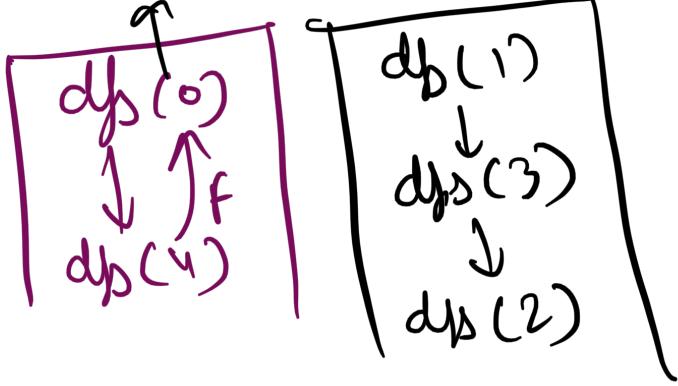
$] \downarrow$

$[$

$T | F | F | F | T$

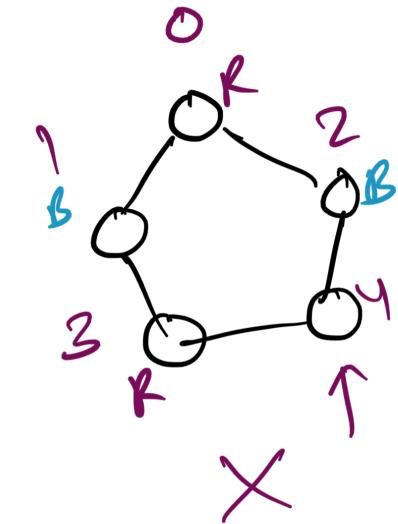
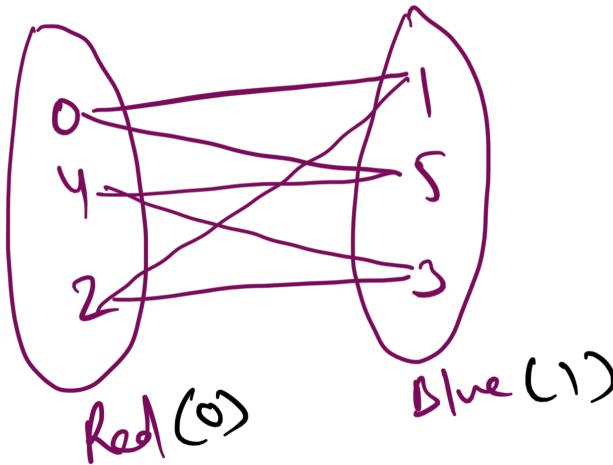
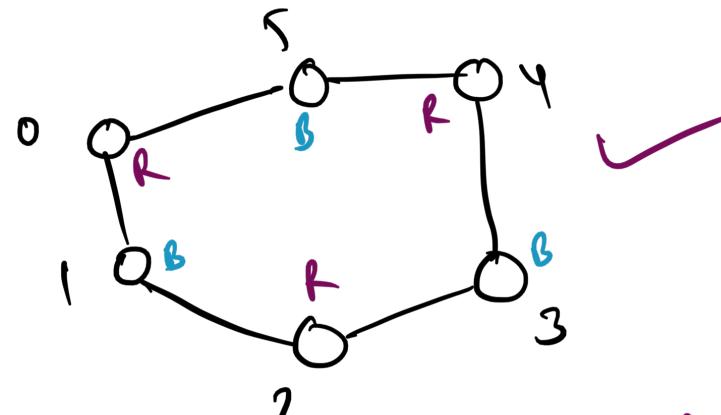
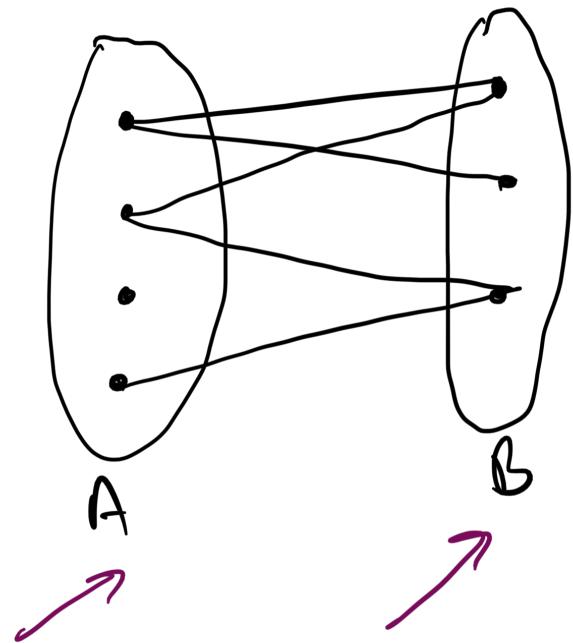
$0 | 1 | 2 | 3 | 4$

$] \uparrow$



0 → 4
 1 → 0, 3
 2 → 1
 3 → 2
 4 →

Bipartite Graph



Color =

-1	-1	-1	-1	-1	-1
0	1	2	3	4	5

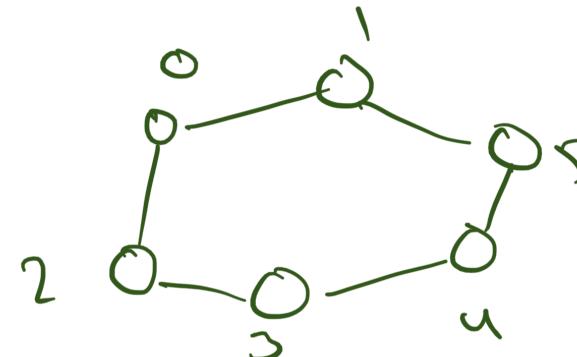
8

	0	1	n	2	3
0	3	3	3	3	
1	3	3^T	0	3	
2	3^T	1	0	3	
3	0	0	2	3	

(i, j)

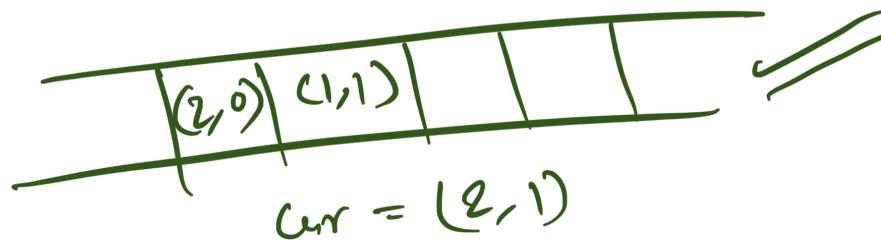
$1 \rightarrow \text{src}$
 $2 \rightarrow \text{dest}$
 $3 \rightarrow \text{blank / Path.}$
 $0 \rightarrow \text{wall}$

class

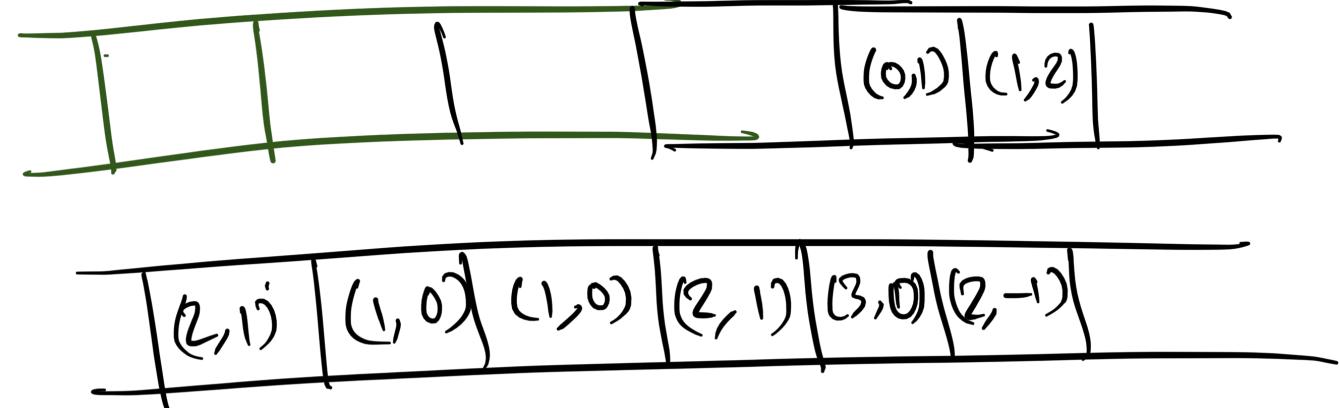


dp X TLE
bfs. ✓ $O(m \times n)$

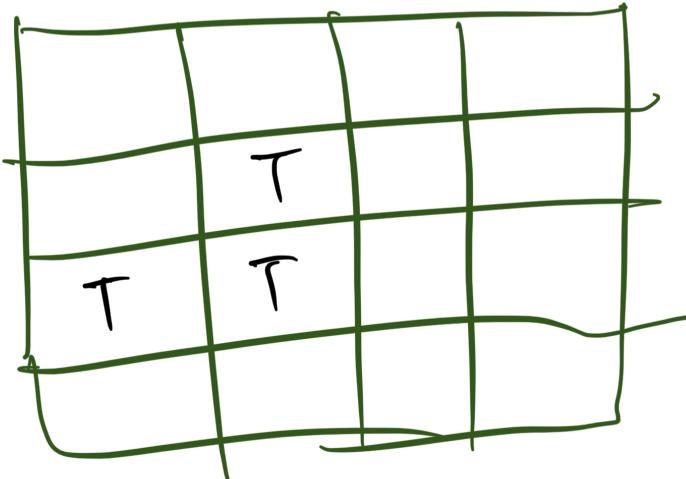
queue



m	0	1	n	2	3
0	3	3	3	3	3
1	3	3	0	3	3
2	3	1	0	3	3
3	0	0	2	3	3



$$cur = (0, 1)$$



Practice Problems

1. Detect cycle in an ~~u~~ directed Graph
 - a. Solution: <https://www.youtube.com/watch?v=GLxfoaZIRqs>
2. Check whether a given graph is Bipartite or not
3. Check whether a path exists

Graphs - III



InDegree of a Directed Graph

Topological Sorting in a Graph

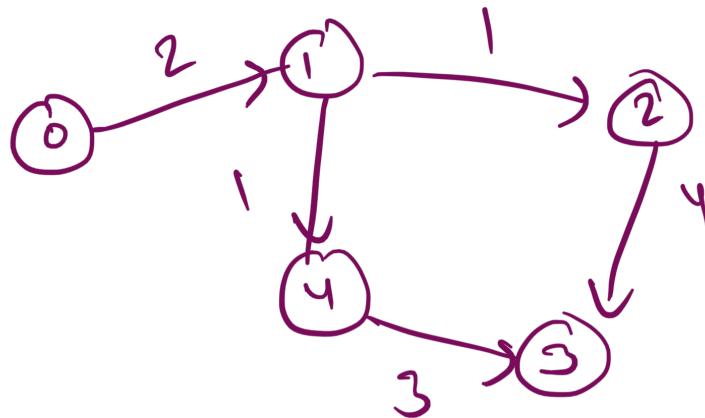


Topological Sort using DFS

Kahn's Algorithm - Topological Sort using BFS

Detect a cycle in a Directed Graph using the Kahn's Algorithm

→ Shortest distance in a DAG



for (Pair neighbor : a.get(4)) {

close Pair S
int v, w
y

