

Graph

debugwithshubham

Contents

0. Introduction
1. All paths from source to target
2. Flood Fill
3. Number of Islands
4. Max Area of the Island
5. Find if path exist in Graph
6. Find the town judge
7. Detect cycle in a Directed Graph
8. Topological Sort
9. Course Schedule
10. Course Schedule II



debugwithshubham



debugwithshubham

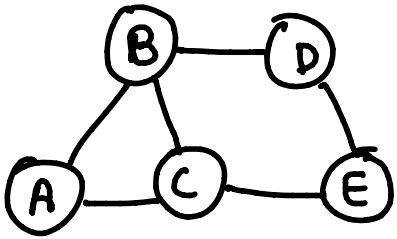


debugwithshubham

Graphs

Graph G is a pair (V, E) where V is set of vertices & E is set of edges. $n = |V|$ & $e = |E|$

Eg



$$V = \{A, B, C, D, E\} \quad n = 5$$

$$E = \{AB, AC, BC, BD, CE, DE\} \quad e = 6$$

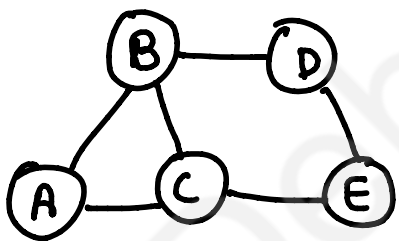
Applications →

Google maps → To find shortest routes

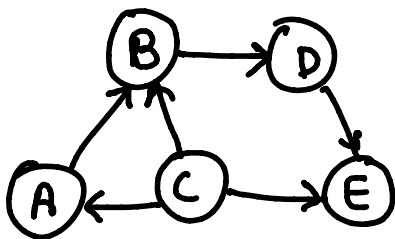
Social network → user, connection
 ↑ ↑
 vertex edge

Types →

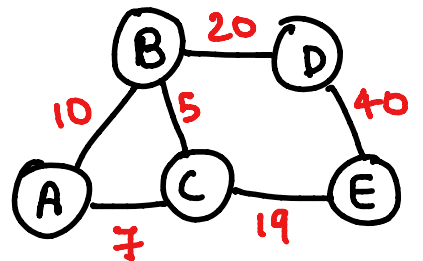
1) Undirected



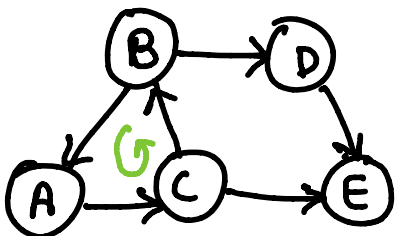
2) Directed



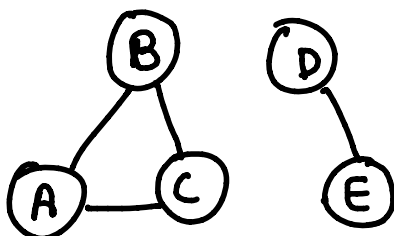
3) Weighted



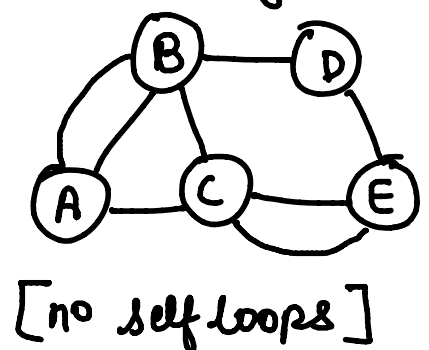
4) Cyclic



5) Disconnected



6) Multigraph

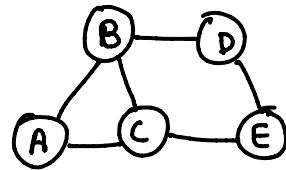


[no self loops]

Graph Traversal

(a) BFS → visit each and every vertex in a defined order.

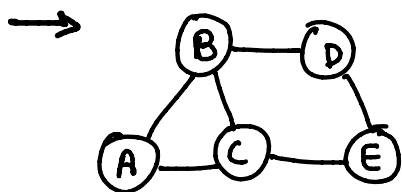
- select node
- visit its unvisited neighbour nodes
- mark it as visited & push into result
- push it into queue
- if no neighbours then pop.
- repeat till queue is empty



queue.

Visited

A	B	C	D	E

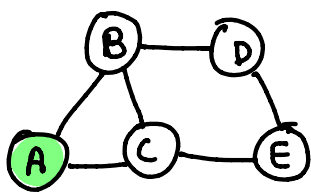


queue.

Visited

A	B	C	D	E

res

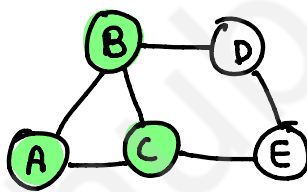


queue.

Visited

A	B	C	D	E

res

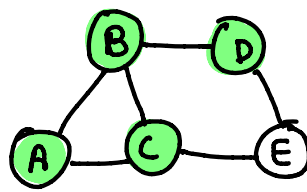


queue.

Visited

A	B	C	D	E

res

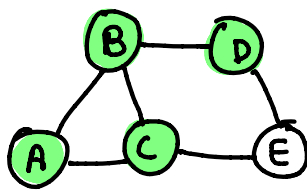


queue.

Visited

A	B	C	D	E

res

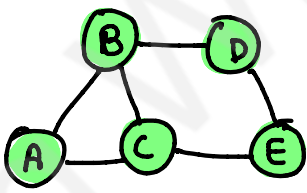


queue.

Visited

A	B	C	D	E

res

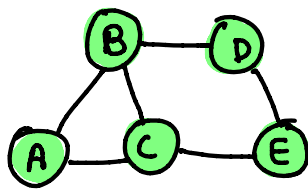


queue.

Visited

A	B	C	D	E

res

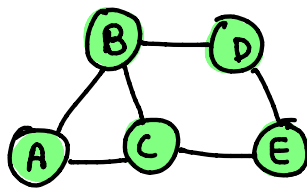


queue.

Visited

A	B	C	D	E

res

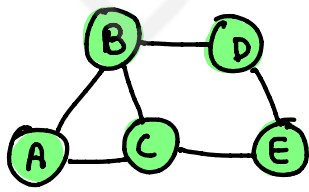


queue.

Visited

A	B	C	D	E

res

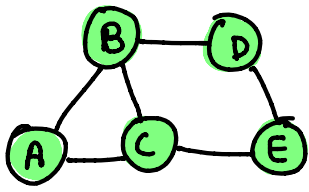


queue.

Visited

A	B	C	D	E

res



queue.

Visited

A	B	C	D	E

res

TC → O(V+E)

SC → O(V)

Return res