***Data II:***

* BFS (single source shortest path for weight 0/1) 🡪 **FIFO (Queue)**
  + Queue
  + Visited (0/1)
  + Exploring
  + Distance (every time add with the previous one) \*
* DFS (cycle finding in a directed graph) 🡪 **LIFO (Stack)**
  + Stack\*
  + Visited (0/1) \*
  + Sequence
  + Exploring
* TopSort 🡪 Directed

🡪 Acyclic

* Incoming edge count
* Queue
* Adjacency list
* Sequence

**\*We can find cycle with TopSort as well**

* Disjoint set
* Minimum spanning tree (mst)
  + Kruskal’s algo
    - U
    - V
    - W
    - Array
    - Total cost

**\*Check whether connected or not**

* + Prim’s algo
    - Priority queue
    - Value
    - Added (0/1)
    - Total cost

**\*in case of MST, Greedy gives the optimal result**

* Single source shortest path
  + Dijkstra (Greedy)
  + Bellman Ford (DP)