Daa Tutorial 5

Ques 1- Difference between Bfs / Dfs. Write application of both.

BFS

it stands for Breadth first search uses queue

data structure for finding the shortest path in level-wise

BFS can be used to find the shortest path, it use to find single source shortest path in unweighted graph.

BFS is better when target is closer from the source

it consider all the neighbour, so it is not suitable for decision tree used in puzzle games

BFS is slower

Time complexity is O(V+E)

Application :- of DFS

→ To detect cycles in graph
→ Path Finding
→ Topological Sorting

DFS

it stands for Depth first search, it uses stack.

it also find the shortest path in depth-wise search

→ In Dfs, we might transverse through more edges to reach a destination vertex from source

Dfs is better when target is far from the source

it consider more suitable for decision tree.

Dfs is faster

it time complexity is also O(V+E)

Application:- Bfs

→ GPS Navigation System
→ Broadcasting
→ P2P to Peer Networking

ques - which data structure to implement and why?

Bfs uses the queue data structure to transverse a graph in a breadthward motion and uses a queue to remember to get the next vertex to start a search, when a dead end occur in any iteration the queue follow the queue concept the vertex that are discovered first will be explored first

Dfs uses the stack data structure to transverse a graph in depthward motion and uses stack to remember to get the next vertex to start a search, when a dead end occur in any iteration

11b- what do you meant by sparse and dense graphs? which representation is better for sparse and dense graphs?

A dense graph is a graph in which the number of edges is close to the maximal number of edges ~~is close to~~ ~~maximum~~ ~~All~~. The every pair of vertices is connected by one edge. The opposite, a graph with only a few edges is a sparse graph. The distinction between sparse and dense graph is rather vague and depends on the context.

The graph density of simple graph is defined to be the ratio of numbers of Edges $|E|$ with respect to the maximum possible edge.

$$D = \frac{|E|}{\binom{|V|}{2}} = \frac{2|E|}{|V|(|V|-1)}$$

11c- Heap data structure can be used to implement priority queue? Name two graphs algorithm where you need to use Priority queue and why?

Binary heap is a Binary data structure. A priority queue can be implemented using an array, a link list, a heap data structure. Among these data structure, heap data structure provides an efficient implementation of priority queue.

Hence, we used heap for implement of priority queue

By inserting an Element into the Priority Queue and delete Element from queue, find Max, Min and Extract Max / Min from the Priority Queue

The graph algorithm uses the priority queue

Dijkstra's Shortest path algorithm

Prim's algorithm

heap sort algorithm

- To detect the cycle in Bfs and Dfs?

In case of Dfs : back edge concept will be used to find out cycles in the graph. If there exist back edge in the graph the it contain a cycle.

In case of Bfs : just the traverse the graph and if any vertex find its adjacent vertex with flag zero then that graph contains a cycle.

Ques - what do you mean by disjoint set data structure? Explain 3 operations along with Example which can be perform on disjoint set

The disjoint set data structure is also known as union - find data structure and merge find set. It is also data structure that contains a collection of disjoint or non-overlapping sets. The disjoint set means that when the set is partitioned into the disjoint subsets. The various operation can be performed on the disjoint subset. In case, we add new set we can merge the set, whether the two element are in the same set or not efficiently.

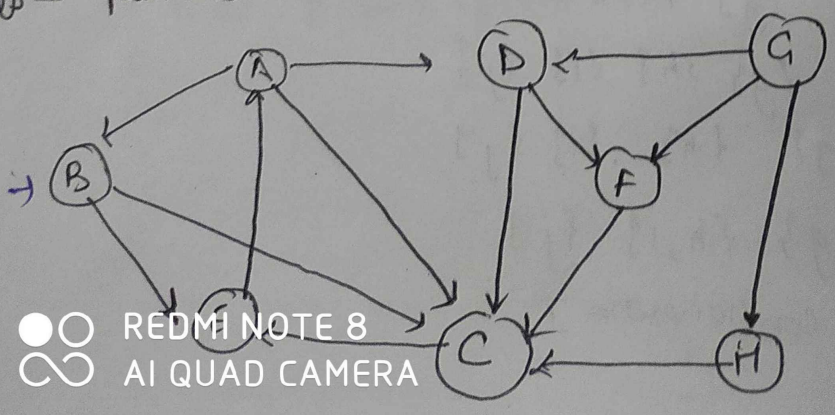
operation on disjoint set

CREATE-SET (x) - Create a new set with one element {x}.

MERGE-SETS (x,y) = merge into one set the set that contains element x and the set that contain element y (x and y are in different sets).

FIND-SET(x) - return the representation or a pointer to the representative of the set that contain

Ques - Run BFS and Dfs on graph :-



BFS

Child	G	H	D	F	C	E	A	B
Parent		G	G	G	H	C	E	A

Path → G → H → C → E → A → B

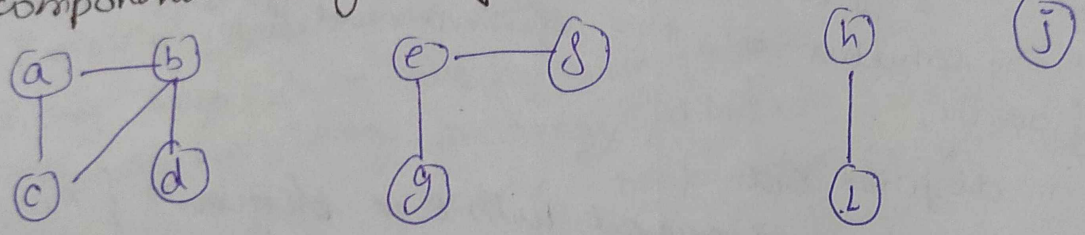
DFS

Nodes visited: G, D, H, F, C, E, A, B

Stack: G, F, C, E, A, B

Path → G → F → C → E → A → B

Ques 7) find out no. of connected component and vertices in each component using disjoint set data structure



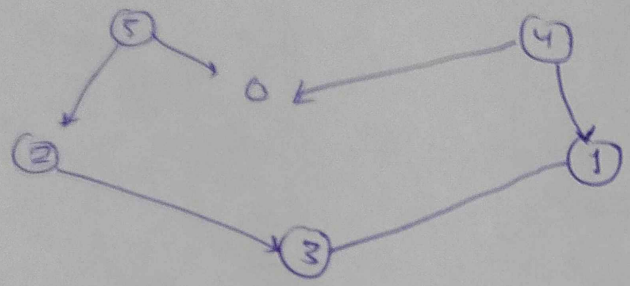
V = {a} {b} {c} {d} {e} {f} {g} {h} {i} {j}

E = {a,b}, {a,c}, {b,c}, {b,d}, {e,f}, {e,g}, {h,i}, {j}

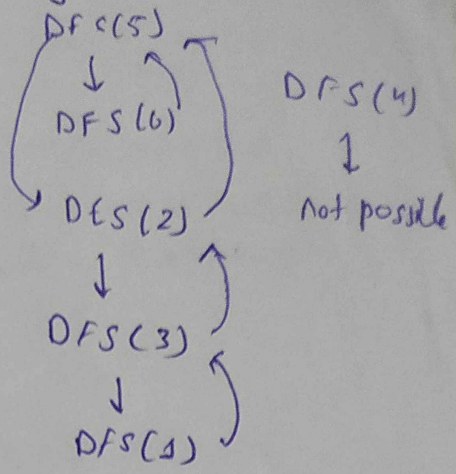
(a,b)	{a,b} {c} {d}, {e}, {f}, {g}, {h}, {i}, {j}
(a,c)	{a,b,c} {d} {e} {f} {g} {h} {i} {j}
(b,c)	{a,b,c,d} {e} {f} {g} {h} {i} {j}
(b,d)	{a,b,c,d} {e,f} {g} {h} {i} {j}
(e,f)	{a,b,c,d} {e,f} {g} {h} {i} {j}
(e,g)	{a,b,c,d} {e,f,g} {h} {i} {j}
(h,i)	{a,b,c,d} {e,f,g} {h,i} {j}

No. of connected component = 3

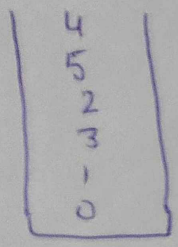
Q-8 Apply topological sort & DFS on graph having vertices from 0 to 5



Applying Topological sort



DFS



Stack

4 → 5 → 2 → 3 → 1 → 0

Q- Differentiate between Min heap and Max heap.

Min heap	max heap
1) In Min-heap, key present at root node must be less than or equal to among keys present at all of its children.	1) In max-heap the key present at root must be greater than or equal to among key present at all of its children
2) The minimum key element is present at the root.	2) The maximum key element is present at the root.
3) It uses ascending priority	3) It uses descending priority
4) The smallest element has priority, while construction of min-heap.	4) The largest element has priority, while construction of max-heap.
5) The smallest element is first to be popped from the heap	5) The largest element is 1st to be popped from the h

DAA TUTORIAL - 6

Name:- Rotf Tikka
Sec:- F
Roll no:- 21
Uni Roll no:- 2046947

Ques 1- What do you mean by minimum spanning tree? What are the applications of MST?

Ans → Minimum spanning Tree is a subset of edge of a connected edge-weighted undirected graph that connects all the vertices together without any cycles & with minimum possible edge weighted

Application:-

Consider n stations are to be linked using a communication network and laying of communication link b/w any two stations involved a cost. The ideal solution would be to extract a subgraph termed as minimum cost spanning tree

i) Designing ~~the~~ LAN

ii) suppose you want to construct highway or railroads spanning tree, then we can use the concept of MST.

iii) Laying pipeline connecting offshore drilling site, refineries & consumer markets

Ques 2 → Analyze time and space complexity of Prim, Krushal, Dijkstra and Bellman Ford Algorithm

Time complexity of Prim's Algorithm $O(|E| \log |V|)$

space complexity of Prim's Algorithm $O|V|$

Krushal's Algorithm

Time complexity $O|E| \log |E|$

Space complexity $O|V|$

Bijkstra's Algorithm

Time complexity $O(V^2)$

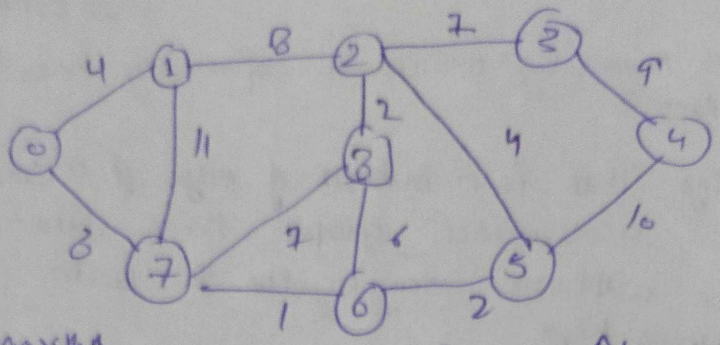
space complexity $O(V^2)$

Bellman Ford's Algorithm

Time complexity $O(VE)$

space complexity $O(E)$

Ques3 - Apply Kruskal and Prism's Algorithm on given graph compute MST and its weight



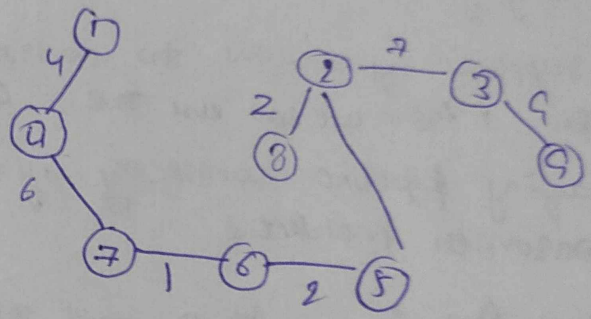
Kruskal's Algorithm

u	v	w
6	7	1
5	6	2
2	8	2
0	1	4
2	5	4
6	8	6
2	3	7
7	8	7
0	7	8
1	2	8
4	3	9
4	5	10
1	7	10
3	5	14

Prim's Algorithm

4 + 8 + 2 + 4 + 2 + 7 + 9 + 3

= 37

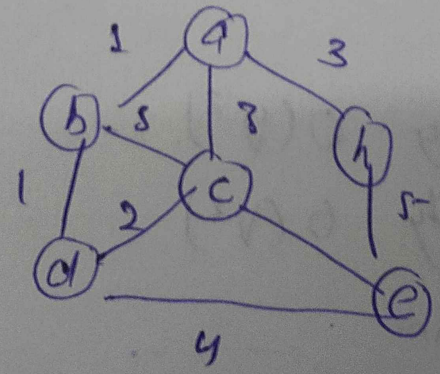


Ques 4 →

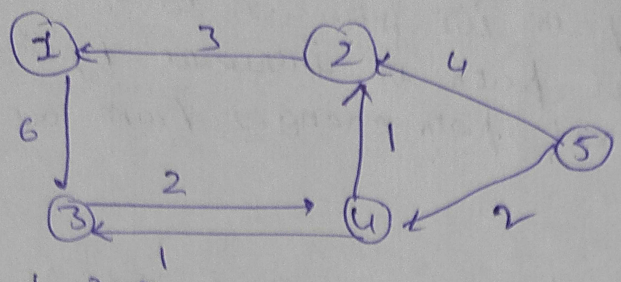
Given a directed weight graph. You are also given the shortest path from a source vertex 's' to a destination vertex 't'. Does the shortest path remain same in following cases:-

i) If weight of every edge is increased by 10 unit

ii) If weight of every edge is increased by 10 unit



Apply all pair shortest path algorithm - Floyd Warshall on below mentioned graph. Also analyse space & time complexity of it



	1	2	3	4	5
1	0	∞	6	3	∞
2	2	0	∞	∞	∞
3	∞	∞	0	2	∞
4	∞	1	1	0	∞
5	∞	4	∞	2	0

	1	2	3	4	5
1	0	∞	6	3	∞
2	2	0	8	5	∞
3	∞	∞	0	2	∞
4	∞	1	1	0	∞
5	∞	4	∞	2	0

	1	2	3	4	5
1	0	∞	6	3	∞
2	2	0	8	5	∞
3	∞	∞	0	2	∞
4	3	1	1	0	∞
5	6	4	12	2	0

	1	2	3	4	5
1	0	∞	6	3	∞
2	2	0	8	5	∞
3	∞	∞	0	2	∞
4	3	1	1	0	∞
5	6	4	12	2	0

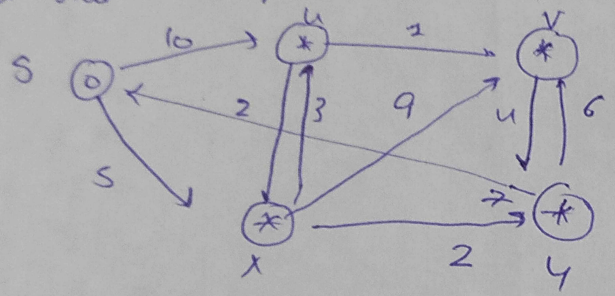
Time complexity
$\rightarrow O(|V|^3)$

Space complexity
$\rightarrow O(|V|^2)$

Ans 1 (i) The shortest path may change. The reason is that there may be different no. of edges in different paths from 's' to 't'
For eg:- Let the shortest path of weight 15 and has edges 5- Let there be another path which increased by 5*10 and becomes 15+50. weight of other path is increased by 2*10 & becomes 25+20, so, the shortest path changes from an increased path with weight as 45.

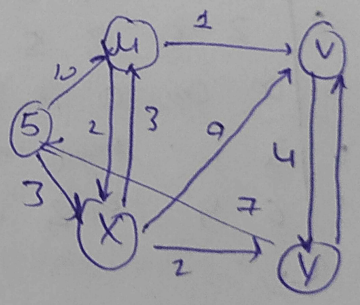
(ii) If we use multiply all edges weight by 10, the shortest path doesn't change. The reason is that weights of all path from 's' to 't' gets multiplied by same unit the number of edges or path doesn't matter.

Que 3 → Apply Dijkstra & Bellman ford algorithm on graph given right side to compute shortest path to all nodes from Node S.

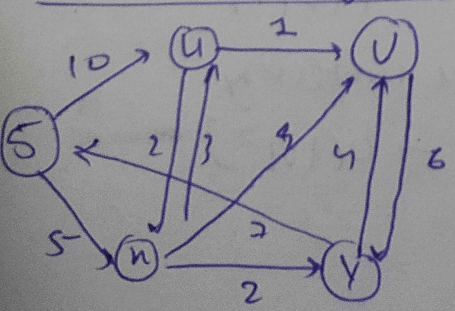


Dijkstra's Algorithm

Node	SHORTEST DIST FROM SOURCE NODE
u	8
x	5
v	9
y	7



Bellford Algorithm



1st	S (0)	u (∞)	v (∞)	x (3)	y (∞)
2nd	S (0)	u (10)	v (11)	x (5)	y (∞)
3rd	S (0)	u (8)	v (14)	x (5)	y (7)
4th	S (0)	u (8)	v (9)	x (5)	y (7)