Placement DSA sheet by Arsh Goyal

**Day-3(‎06-‎05-‎2022)**

**Q-1=> BFS of graph (**[**https://practice.geeksforgeeks.org/problems/bfs-traversal-of-graph/1**](https://practice.geeksforgeeks.org/problems/bfs-traversal-of-graph/1)**)**

hint\_1=> we have to move step by step means layer by layer and keep visited vector updated and use queue to store the vertexes.

**Tips - 1 =>** when we travese graph through bfs then we reach at any vertex in minimum time from root vertex. So we can use this aproach to find the shortest path b/w two vertex. We also find the shortest path using dfs but it will take more time than this aproach.

**Q-2=> DFS of graph (**[**https://practice.geeksforgeeks.org/problems/depth-first-traversal-for-a-graph/1/#**](https://practice.geeksforgeeks.org/problems/depth-first-traversal-for-a-graph/1/#)**)**

hint\_1=> in this question we have to move recursivily first we complete each of the element conected with other and then we move to other.

**Q-3=> Detect cycle in an undirected graph (**[**https://practice.geeksforgeeks.org/problems/detect-cycle-in-an-undirected-graph/1#**](https://practice.geeksforgeeks.org/problems/detect-cycle-in-an-undirected-graph/1#)**)**

hint\_1=> hame ye dhyan rakhna hai ki ye undirected graph hai to ham apne just last wale vertex pe nahi ja sakte hai iske liye hame ek pre data ko fun. ke sath hi bhejna hoga aur ham dfs ke samay pre ke alawa baki sab me jayenge. baki sab kuchh dfs hi hai.

**Q-4=> Detect cycle in an directed graph (**[**https://www.geeksforgeeks.org/detect-cycle-in-a-graph/**](https://www.geeksforgeeks.org/detect-cycle-in-a-graph/)

hint\_1=> iss wale circle detection me hame ye dhyan rakhna hai ki ham jab dfs karte hue deep ja rahe ho to agar kahi pe visited 1 milta hai to sirf tabhi 1 return karna hai jab uss node ko ham abhi tak jitne depth pe ho waha tak usko traverse kar ke aaye ho isiliye ham ek aur check vector lekar chalenge aur jab deep jayenge to usko call kar denge aur jab bahar aayenge to usko delete kar denge and time complexity ke liye hame agara kisi round ko check kar chuke hai to uss round ke pahle hi return 0 kar dena hai;

**Q-5=> Topological sort (**[**https://practice.geeksforgeeks.org/problems/topological-sort/1**](https://practice.geeksforgeeks.org/problems/topological-sort/1)**)**

hint\_1=> in this question we have to use stack to track the item in deep jab bhi ham deep jaye aur jab bhi ham sabse deepest place se nikalne lage to uss ele ko stack me push kar de phir last me stack me se top se nikal kar usko vector me dal lenge.

**kahn's algorithem =>** jab ham topological short ko bfs ka use karke karte hai to usi ko kahn's algorithem kahte hai. isme hame ek aur data structure ka use hoga jisme ham ye store karenge ki kis node pe kitne path aa rhe hai aur jis node pe koi bhi path nahi aa raha hoga wo obius si bat hai ki pahle hi rahega isiliye use ham queue me store kar lenge aur jab bhi ham isko pop kar ke iske agal bagal walo ko traverse karenge to unke incoming path ko - kar denge aur agar wo 0 hai to queue me le lenge;

**kahn's algorithem =>** isi ka use kar ke ham directed graph me cycle bhi detect kar sakte hai using bfs (mens agar koi puchhe ki bfs se cycle kaise find karenge directed graph me to ham simple kahn's algorithem ka use karke topological sort vector nikalenge agar hame valid vector mila matlab cycle nahi hai kyuki topological sort directed acyclic graph me hi hota hai cyclic me ye nahi hota ) aur validataion check karne ke liye simple vector ke length ko number of node se compare kar lenge agar = hua matlab valid hai aur nahi to invalid;

**Q-6=> Shortest path in an undirected unweighted graph (**[**https://www.codingninjas.com/codestudio/problems/shortest-path-in-an-unweighted-graph\_981297?leftPanelTab=0 HYPERLINK "https://www.codingninjas.com/codestudio/problems/shortest-path-in-an-unweighted-graph\_981297?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"& HYPERLINK "https://www.codingninjas.com/codestudio/problems/shortest-path-in-an-unweighted-graph\_981297?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"utm\_source=youtube HYPERLINK "https://www.codingninjas.com/codestudio/problems/shortest-path-in-an-unweighted-graph\_981297?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"& HYPERLINK "https://www.codingninjas.com/codestudio/problems/shortest-path-in-an-unweighted-graph\_981297?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"utm\_medium=affiliate HYPERLINK "https://www.codingninjas.com/codestudio/problems/shortest-path-in-an-unweighted-graph\_981297?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"& HYPERLINK "https://www.codingninjas.com/codestudio/problems/shortest-path-in-an-unweighted-graph\_981297?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"utm\_campaign=Lovebabbar**](https://www.codingninjas.com/codestudio/problems/shortest-path-in-an-unweighted-graph_981297?leftPanelTab=0&utm_source=youtube&utm_medium=affiliate&utm_campaign=Lovebabbar)**)**

hint\_1=> hame isme bfs ka use karna hai bas yahi change karna hai ki jab bhi ham kisi vertex pe jaye to use vertex ke parent ko store kar lena hai isse ham reverse direction me traverse kar ke pta laga sakte hai shortest path kya hai (**tips 1**);

hint\_2=> ham isme dfs ka bhi use kar sakte hai uske liye hame do array aur ek lenth ko extra bnana hai ek array me temprary path hoga jab bhi ham deep jayenge to usme vertex ko dal denge aur jab bahar ayenge to nikal lenge. Aur jab ham apne target wale vertex pe honge to uspe ham length ko compare karenge aur agar length > hoga to main array me temprary wale ko store kar ke length ko change kar lenge.(see the tle submision of above question);

**Q-6=> Shortest path in an undirected weighted graph (**[**https://www.codingninjas.com/codestudio/problems/dijkstra-s-shortest-path\_920469?leftPanelTab=0 HYPERLINK "https://www.codingninjas.com/codestudio/problems/dijkstra-s-shortest-path\_920469?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"& HYPERLINK "https://www.codingninjas.com/codestudio/problems/dijkstra-s-shortest-path\_920469?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"utm\_source=youtube HYPERLINK "https://www.codingninjas.com/codestudio/problems/dijkstra-s-shortest-path\_920469?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"& HYPERLINK "https://www.codingninjas.com/codestudio/problems/dijkstra-s-shortest-path\_920469?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"utm\_medium=affiliate HYPERLINK "https://www.codingninjas.com/codestudio/problems/dijkstra-s-shortest-path\_920469?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"& HYPERLINK "https://www.codingninjas.com/codestudio/problems/dijkstra-s-shortest-path\_920469?leftPanelTab=0&utm\_source=youtube&utm\_medium=affiliate&utm\_campaign=Lovebabbar"utm\_campaign=Lovebabbar**](https://www.codingninjas.com/codestudio/problems/dijkstra-s-shortest-path_920469?leftPanelTab=0&utm_source=youtube&utm_medium=affiliate&utm_campaign=Lovebabbar)**)**

**Dijkstra's algorithem =>** iss algorithem ka use ham undirected wieghted graph me shortest path nikalne ke lihye karte hai isme sare weights +ve hone chahiye tabhi ye kam karega.

isme ham ek set ya phir priority queue lete hai aur source node ke distance =0 aur node ko insert kar dete hai iske baad ham apne ans vector ko define karte hai infinite ke sath aur iske bad ham set se minimum distance wala node nikalte jayenge aur phir us node ke sare neighour pe jake unpe check karenge ki wo current dist. se < hai ki nahi agar < hai to set mese agar unka pahle se koi hoga to usko nikal kar naye wale ko enter kar denge;

**Q-6=> Shortest path in a directed acyclic graph(weighted ) (**[**https://ideone.com/nA6Np4**](https://ideone.com/nA6Np4)**)**

hint\_1=> iss question ke liye apne pass bhut se alborithm hai jaise bellmon ford o(ve), dijkstra's o(e+vlogv) but ye sab efficient aproach nahi hai agar ek vertex se sabhi vertex ka shortest distance puchha hai to iske liye ham pahle **topological short** wale vector ko nikalna hoga iske baad ham vector ke sabse bahar wale vertex se chalu karenge aur andar tak jayenge aur jab bhi koi int\_max ke == nahi hoga uske liye phir adj me ghus kar uske sare adjacent pe ja ke check karna hai aur jo bhi uska weight hai usme current wale se compare karke usko add kar dena hai agar wo pahle wale se kam hai to;