|       | 21/06/2023 ulgarie - painturbre anna e America                             |
|-------|--|
|       |  |
|       | Topological Sort   |
|       | Linear ordering of vertices such that for every                            |
|       | edge UTV, u comes before v in that ordering.                               |
|       | Jessie Villiam Orgening.   |
|       | <u>(1)</u>   |
|       |  |
|       | $(2) \longrightarrow (3)$  |
|       | 1-13 )   |
|       | 1-3 ? Adjacency list   |
|       | melate 2500 grania manej Laternano sistem entre                            |
|       | Valid totalogical sort = 1-2-3   |
|       | Valid topological sort ⇒ 1 → 2 → 3<br>Invalid topological sort ⇒ 2 → 1 → 3 |
|       | =nvalid copological solution   |
| Vote- | Nodes with no dependency are printed.                                      |
|       | Topological sort can be applied only on DAG                                |
|       | l'e directed acyclic graph.  |
| V (   | 2 done do male or or or or or doe doe or the size of the order             |
| Exy   | CANAL & CANAL K HOUR STUDGE STORTER & COOCH                                |
| Jal I |  |
|       | 1  |
|       | (2)  |
|       | dfs(0)   |
|       | the transformation of the  |
|       | dfs(1)   |
|       | 4  |
|       | dfs(2)   |
|       |  |
|       | $dfs(3) \longrightarrow dfs(5)$  |
|       | 100(2)   |
| )     | $dfs(4) \rightarrow dfs(6) \rightarrow dfs(7)$                             |

|              | node   |           | le returning, simply push that        |  |  |  |  |  |
|--------------|--|-----------|---------------------------------------|--|--|--|--|--|
| $\parallel$  |  |           | Japan Sant.                           |  |  |  |  |  |
| 4            | t feet   | 0         | but and on the primative rusting      |  |  |  |  |  |
| 1            | 45.1 to  | 4 7       | NESSTANDA DESTRUCTOR LESS OF          |  |  |  |  |  |
| $\downarrow$ |  | 2         |                                       |  |  |  |  |  |
| #            |  | 3         |                                       |  |  |  |  |  |
| #            |  | 5.        |                                       |  |  |  |  |  |
| $\parallel$  |  | 4         |                                       |  |  |  |  |  |
| $\parallel$  |  | 6         | <u> </u>                              |  |  |  |  |  |
| $\parallel$  |  | 7         |                                       |  |  |  |  |  |
| $\parallel$  | Now J  | Petch     | elements from stack and we will       |  |  |  |  |  |
| $\parallel$  | get a valid topological Sort.                      |           |                                       |  |  |  |  |  |
| $\parallel$  | 0-1-2-3-5-4-6-7                                    |           |                                       |  |  |  |  |  |
| $\parallel$  |  |           |                                       |  |  |  |  |  |
| $\parallel$  | Cada   | 3-1 9 186 | of Nodes with no dependently use b    |  |  |  |  |  |
|              | Loge   | in the b  | Topological sort son but wholish      |  |  |  |  |  |
| $\parallel$  | Void   | . 1 - 0   | - adding al property sign             |  |  |  |  |  |
|              | bool >   | 0 1/10 1  | ort (int src, unordered -map Kint     |  |  |  |  |  |
| ╢            | DOOX / 1   | ~ VIS1-   | ted, Stack (int) & ans) {             |  |  |  |  |  |
|              | Visited [src] = true;                              |           |                                       |  |  |  |  |  |
|              | for (out & place of the form)                      |           |                                       |  |  |  |  |  |
|              | for (auto nbr: adylist[src]) {  if (Ivisited[nbr]) |           |                                       |  |  |  |  |  |
|              | tobo Soxt (nb.                                     |           |                                       |  |  |  |  |  |
| - 11         | topoSort (nbr, visited, ans);                      |           |                                       |  |  |  |  |  |
| 1            | // W   | hile r    | eturning. bush nada in the            |  |  |  |  |  |
|              |  | bush      | eturning, push node in the stack      |  |  |  |  |  |
|              | ans  |           |                                       |  |  |  |  |  |
|              | 3<br>3   | 1         |                                       |  |  |  |  |  |
|              | 3<br>3   |           |                                       |  |  |  |  |  |
|              | 3  |           | ort using BFS. vn as Kahn's algorithm |  |  |  |  |  |

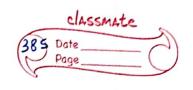
Scarineu with cam

|      | 383 Date Page  |
|------|--|
|      |  |
|      |  |
|      |  |
|      | (2)  |
|      |  |
|      | (3) $(4)$  |
|      |  |
|      | (5)  |
| 100  |  |
| 1000 | (7)  |
|      | The nodes have indegree = 0, simply bush in  |
|      | quelle.  |
|      | The Late of the Control of the Contr |
|      | que + (03 (Remove O from graph)  |
|      | quelle 7 f O = 13 (Remove 1 lucas accept)  |
|      | que + {0,13 (Remove 1 from graph)  que + {0,1,23 (Remove 2 from graph)   |
|      | que 1 {0,1,2,4,33 (Remove 4 & 3 from   |
|      | graph)   |
|      | queue + {0,1,2,4,3,53 (Remove 5 from graph)  |
|      | Queue 7 {0,1,2,4,3,5,6,73 (Remove 6 and 7  |
|      | from graph)  |
|      |  |
|      | Hence topological sort is  |
|      | Hence topological sort is<br>071727473757677.  |
|      |  |
|      | Code   |
|      | Veter with the net required in case of backers   |
|      | Void topo Sort (int n, vector (int) & ans) {   |
|      | quite The 200  |
|      | unordered map (int) indegree;  |
|      | L'alculate indegree<br>For (auto i : adjlist) {  |
|      | for (auto i: adylist) {  |

Scarineu with Carn

| -     |   |
|-------|---|
|       | int syc = x first;  |
|       | for (auto nbr: 1 second) {  |
|       | indegree (nbr]++;   |
|       | 3   |
|       | 3   |
|       | // Put nodes in queue having Indegree = 0   |
|       | for (int $i = 0$ ) $i < n > i + + $ ) {   |
|       | if (indegree [i] = =0)  |
|       | q.push(i);  |
|       | 3   |
| 17    | MBFS logic  |
|       | while (!q.empty())?   |
|       | int front Node = q.front();   |
|       | 9. pop();   |
|       | ans. push_back (front Node)   |
|       | for (auto nbr: adjlist [front Node]){   |
|       | To remove nodes from graph  |
|       | indegree [nbr] i  |
|       | if (indexes [nbx] = = 0)  |
|       | //Check for O indegree again  if (indegree [nbr] = = 0) {  q. push (nbr);   |
|       | 4 9. PUSIL (1) DY)  |
|       | 2 Entrop languaged model  |
|       | 3 .16368646666  |
|       | 3   |
|       | 91.0  |
| Note. | visited is not required in case of connected  |
|       | graphs as we are checking for indearce before   |
|       | & then pushing in the queue. Visited is not   |
|       | required for disconnected graphs also as  |
|       | we have traversed the full graph and  |
|       | Visited is not required in case of connected graphs as we are checking for indegree before & then pushing in the queue. Visited is not required for disconnected graphs also as we have traversed the full graph and calculated the indegree. |
|       | Scarnieu wini odl   |
|       | Scarnica with oar   |

Scarineu with CdM



|        | 1 .  | ١.  |            |    |      |      |    |         |     |
|--------|------|-----|------------|----|------|------|----|---------|-----|
| Why    | topo | 100 | gic        | al | sort | cant | be | applied | on  |
| 0      | '    | . ( | ,          | _  |      |      |    | APPIICA | OIL |
| cyclic | a910 | αЫ  | ነ <u>ይ</u> | 2  |      |      |    | · ·     |     |
| 2      | 9.   |     | •          | 4  |      |      |    |         |     |
| _      |      |     | •          | 1  |      |      |    |         |     |

indeg = 1

 $\frac{B}{\text{indeg}=1}$ 

As here indeg = 1 for all nodes & indeg = 0 for no node and hence we are not able to print all nodes of topological sort.

Cycle detection using BFS.

If we are able to find valid topological sort, then cycle is absent. If no valid topological sort, sort, then simply say cycle is present.

main() ->

if (ans. size() = =n) graph

cout <<"(No cycle");

else

cout << "cycle present";

Scanned with Cam