# L21 - Topological Sort/Job Sequences

Tuesday, June 9, 2020 8:20 AM

An application of Directed Acyclic Graph (DAG) i.e. a directed graph without cycle.

Topological Sorting of a DAG is Linear Ordering of its vertices such that for an edge (u, v), the vertex u appears before vertex v.

TOPOLOGICAL-SORT($G$)
1 call DFS($G$) to compute finishing times $v.f$ for each vertex $v$
2 as each vertex is finished, insert it onto the front of a linked list
3 **return** the linked list of vertices

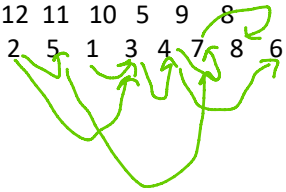
$O(|V| + |E|)$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Predecessor: | Nil | 1 | 3 | 4 | 4 | 7 | Nil | 2 |
| Node: | 1 | 3 | 4 | 6 | 7 | 8 | 2 | 5 |
| Discovery Time: | 1 | 2 | 3 | 4 | 6 | 7 | 13 | 14 |
| Finishing Time: | 12 | 11 | 10 | 5 | 9 | 8 | 16 | 15 |

DFS(G)
1. for each vertex k ∈ G.V
2. u. *color* = WHITE
3. k.$\pi$ = NIL
4. Linked_list LL
5. *time* = 0
6. for each vertex k ∈ G.V
7. if k. *color* == WHITE
8. DFS-VISIT(G; k)

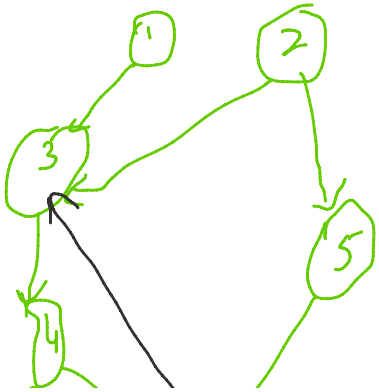
DFS-VISIT(G; k)
1. *time* = *time* + 1 // white vertex u has just been discovered
2. k. *d* = *time* // k. d means the time when vertex k was first **discovered**.
3. k. *color* = GRAY
4. for each l ∈ G.Adj[k] // explore edge (k; l)
5. if l. *color* == WHITE
6. l.$\pi$ = k
7. DFS-VISIT(G; l)
8. k. *color* = BLACK // blacken k; it is finished
9. *time* = *time* + 1

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Predecessor: | Nil | 1 | 3 | 4 | 4 | 7 | | |
| Nodes: | 1 | 3 | 4 | 6 | 7 | 8 | | |
| Discover Time: | 1 | 2 | 3 | 4 | 6 | 7 | | |
| Finishing Time: | 12 | 11 | 10 | 5 | 9 | 8 | | |
| LL : | 2 | 5 | 1 | 3 | 4 | 7 | 8 | 6 |



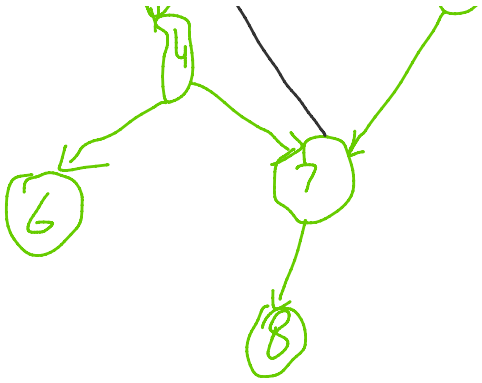
Adjacency List Representation

| | | |
|---|---|---|
| 1 | 3 | |
| 2 | 3 | 5 |
| 3 | 4 | |
| 4 | 6 | 7 |
| 5 | 7 | |
| 6 | | |
| 7 | 8 | |
| 8 | | |



7. DFS-VISIT(G; l)
8. k. *color* = BLACK // blacken k; it is finished
9. *time* = *time* + 1
10. k. *f* = *time* *//finish time*
11. LL.Insert_at_Head(k)

Time Complexity of Topological Sorting O(|V|+|E|)




Predecessor:
Node:
Discovery Time:
Finish Time:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Predecessor: | Nil | 3 | 4 | 4 | 7 | Nil | 1 | 2 |
| Node: | 3 | 4 | 6 | 7 | 8 | 1 | 2 | 5 |
| Discovery Time: | 1 | 2 | 3 | 5 | 6 | 11 | 13 | 14 |
| Finish Time: | 10 | 9 | 4 | 8 | 7 | 12 | 16 | 15 |

2 5 1 3 4 7 8 6