Class 94

25 May 2023 21:08

Class 94

Graphs

Detect cycle in undirected graph → DFS
→ BFS

Directed Graph → using DFS

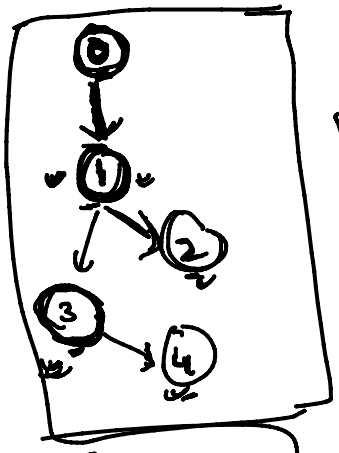
Topological Sorting

(DAG) → Directed Acyclic Graph → Directed
→ Acyclic (No cycle)

Top sort → (u) → (v)

Show Dependency

(A) → (B)



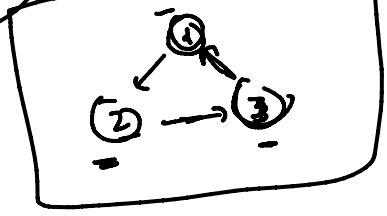
Topsort

0 1 2 3 4

0 1 3 4 2

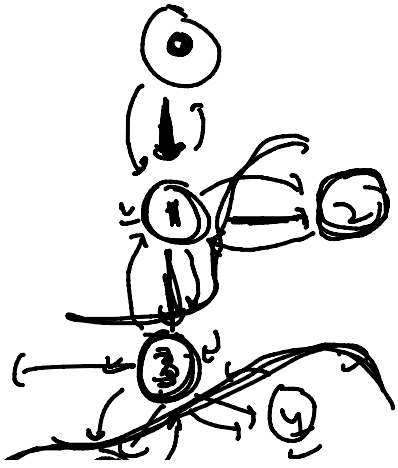
Dependency Graph

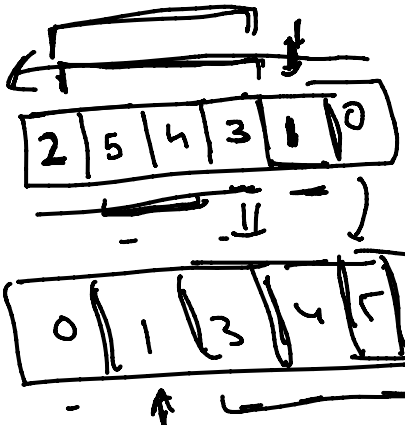
5 sorted

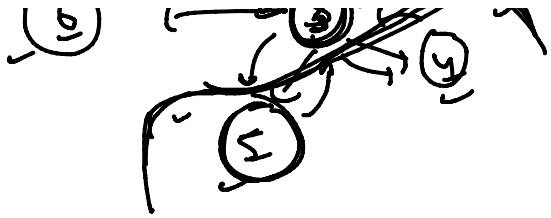


Top sort → Directed Acyclic graph

(u) → (v)

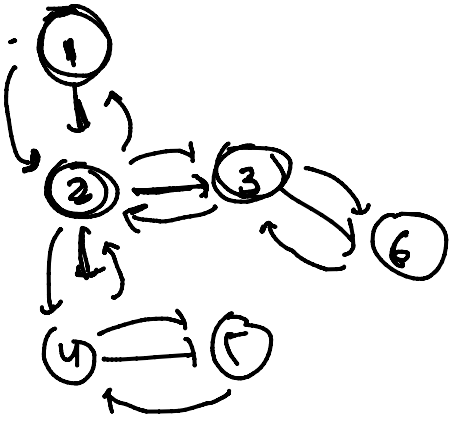


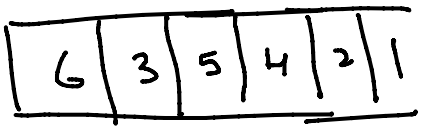


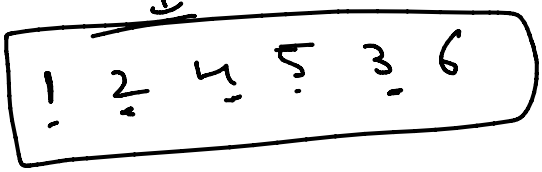


```
void topSort ( vector<vector<int>> adj, vector<int> &vis,
               int node, vector<int> &temp )
{
    vis[node] = 1;

    for (auto i : adj[node])
    {
        if(!vis[i])
        {
            topSort ( adj, vis, i, temp);
        }
    }

    temp.push_back (node);
}
```

```
reverse ( temp.begin(), temp.end());
```

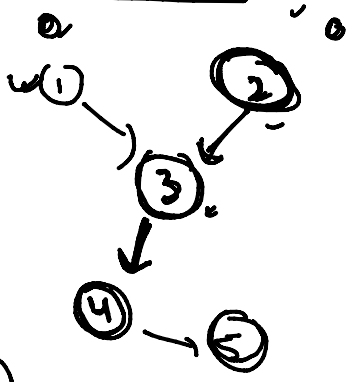


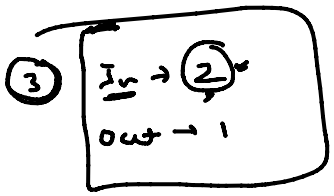


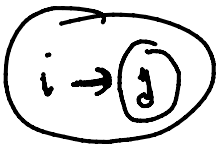


Top Sort using BFS

Indegree or Outdegree







KHANS Algorithm

```
vector<int>indegree(n,0);
    For( i=0 ; i<n ; i++)
    { For( auto j: adj[i])
      {
          indegree[j]++;
      }
    }

queue<int>q;
For( i=0; i<n; i++)
{ if(indegree[i]==0)
   { q.push(i);
   }
}

vector<int> top;
while(!q.empty())
{ auto tp= q.front();
   top.push_back(tp);
   q.pop();
```

For(auto i : adj[top])

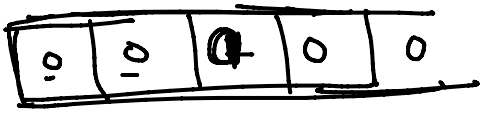
↓ indegree[i]--;

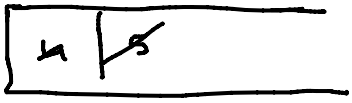
if(indegree[i] == 0)

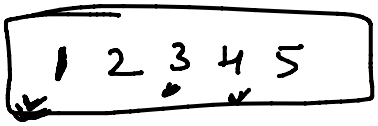
{ q.push(i);

}

}

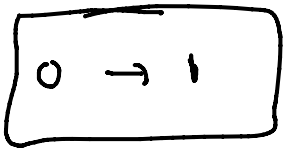






bi → ai

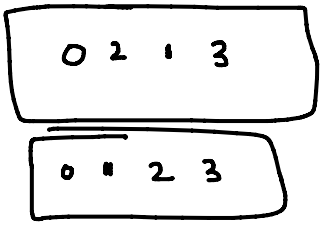
(1,0)

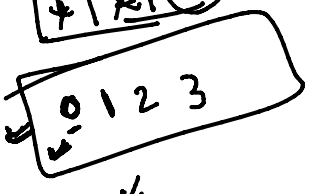


0,1

(1,0) (2,0) (3,1) (3,2)

0 ! 3





2

DAG

Detect cycle in Directed graph.

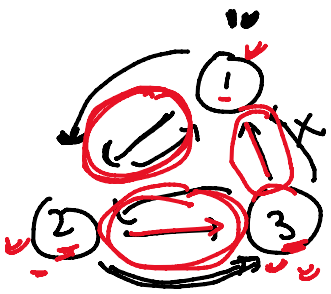
→ Top sort

Top sort

Detect cycle in directed graph.

Top sort

(u) → (v)



(1) → (2)

(3) → (1)

3 2 1 ⇒ 1 2 3

1 → 0
2 → 1
3 → 2

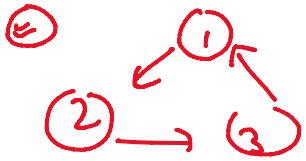
cycle

Top order

Not correct

```
vector<int> topo;
map<int,int> ma;

For(i=0; i<n; i++)
{ ma[topo[i]] = i; }

For ( i=0; i<n; i++ )          i → 1.
{ For (auto j : adj[i])
  { if (ma[i] > ma[j])
      { //cycle detected return false; }
  }
}
return true;
```

Detect cycle using KAHN'S Algorithm



→ [ ]

topSort.size() != No of Node

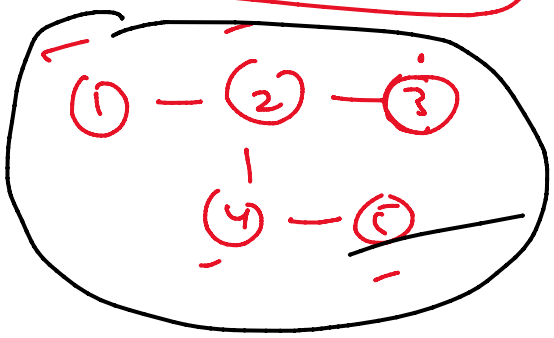
top sort . size() != No of Node

Detect cycle in

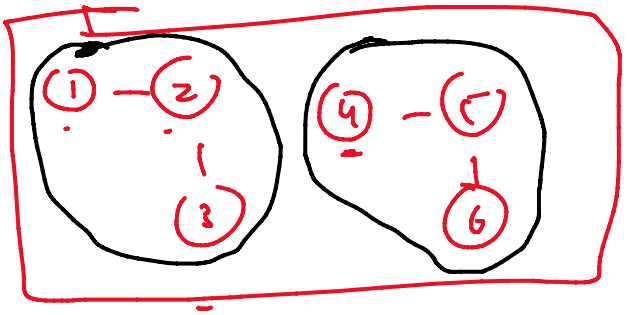
- Undirected
  - DFS
  - BFS
- Directed
  - DFS (coul start)
  - DFS top sort
  - BFS (KAHN)

Components in Graph

Connected component

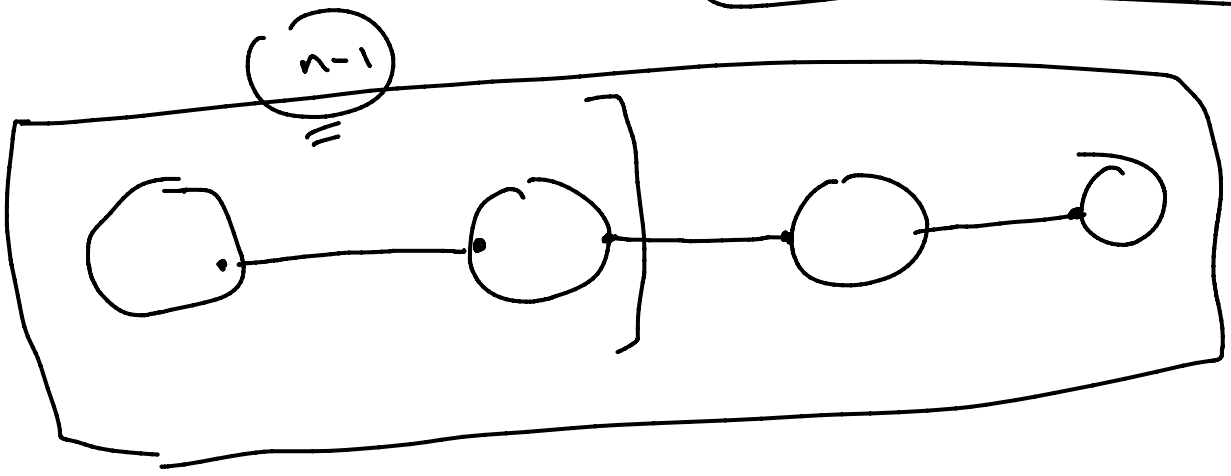


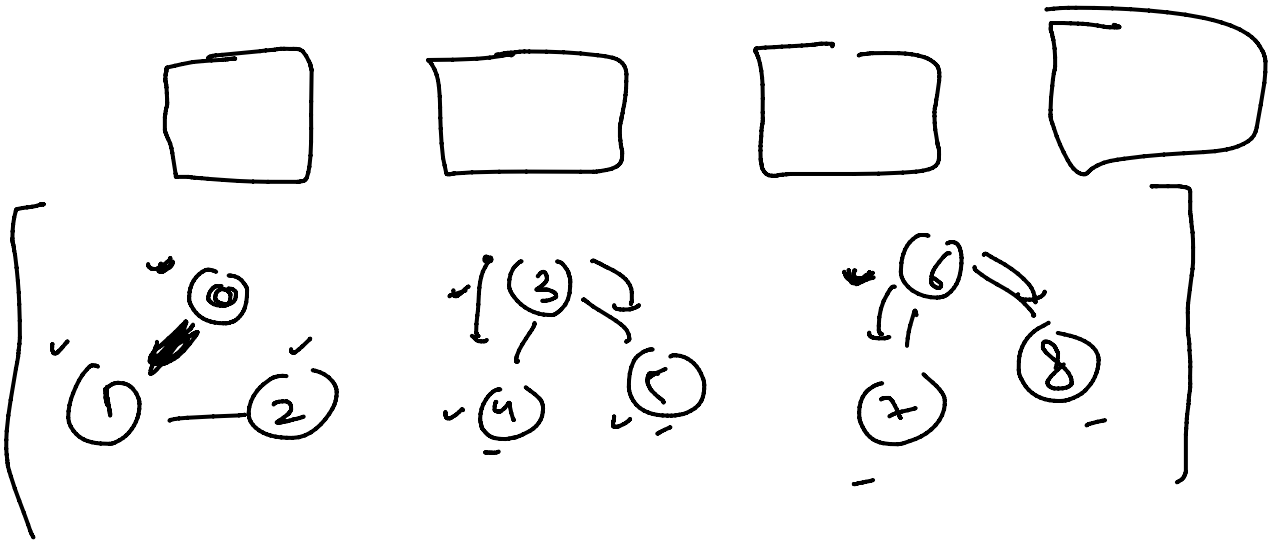
unconnected comp



Total connected component

(n) Connected component → connected component





Vector <int> vis;

For (i = 0; i < n; i++)
{
  if (vis[i] == 0)
  {
    dfs(i, adj, vis);
  }
}

0, 1, 2 ⇐ vis[i] = 1

3, 4, 5

Shortest Path Algorithm → Dijkstra, Bellman Ford, BFS,

Articulation Point / Bridges

Euler circuit