2/28 Topo Sort Cont.

Wednesday, February 28, 2018 4:46 PM

1. theory
2. Topo Sort
3. code

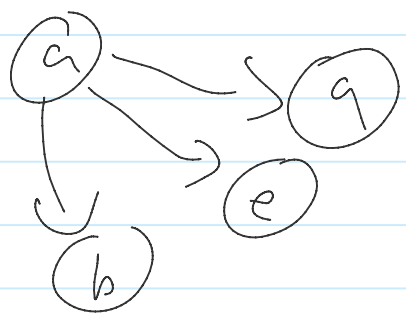
If G is a DAG then there exist a Topo Sort of G.

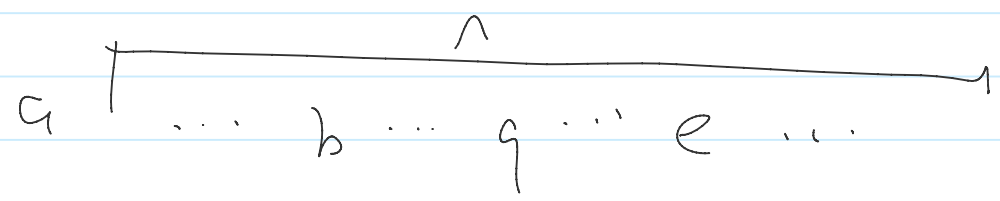
Base case $|V(G)| = 1$

Simply use the node as the Toposort

Suppose we can toposort any graph with n vertices.

Then consider a graph with $n+1$ vertices.





Suppose no Node without incoming edges exist. Then select a random node traverse one of its incoming edges to the "source" vertex

After traversing $n+2$ edges by PHP exists a duplicate node ($n+2$ nodes in list $n+1$ nodes in graph G). This duplicate allows us to form a cycle. Thus

P+/P if you have n+1 pigeons
and n holes ∃ a pigeonhole with
atleast two pigeons.

allows us to form a cycle. Thus we have a contradiction and there must exist a node with no incoming edges. Remove node do Topo sort on N remaining nodes. By Principle of Induction ▨

```
Topo Sort DFS (G)
    mark all nodes unvisited
    Store in degree for each node in array
    for (Node n in G)
        if (!visited(n) and in degree is 0)
            DFS(n, G)
```

```
DFS(cur, G){
    print(cur)
    visit[cur] = true
    for (Edge e where cur is source e){
        decrement indegree of endpoint of e;
        if (indegree of endpoint of e is 0){
            DFS(endpoint of e, G)
        }
    }
}
```