Lecture 6

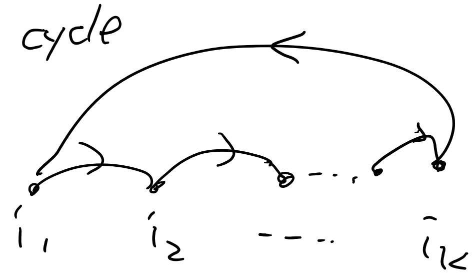
Monday, October 8, 2018 1:18 PM

Thm G has a topo order $\Leftrightarrow$ G is a DAG

Lemma 1 G has topo order $\Rightarrow$ G is a DAG

Proof by contradiction Suppose G has topo order v_i

Suppose G is not a DAG, there is a directed cycle



Let i_1 is the lowest index in the cycle.

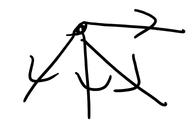
Cycle $\Rightarrow$ there is edge (v_{i_k}, v_{i_1})

Since i_1 is lowest, $i_k > i_1$

this contradicts to the fact v is a topo order.

Lemma 2a If G is a DAG, it has ~~the~~ a source node

(source node = node with no incoming edge)



Proof by contradiction

Algo:

Start at any node v

while there is an incoming edge (u,v) from v

$v \leftarrow u$

Output v

(used DAG)

Termination:

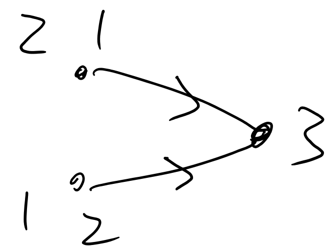
No node is repeated in the loop because no directed cycle.

Hence, it takes $O(n)$ time.

Correctness: v is source because of the loop condition.

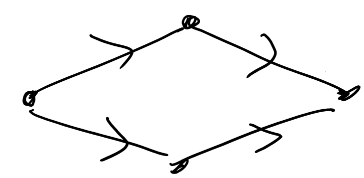
Correctness: v is source because of the loop condition.

Lemma 2b. G is a DAG $\Rightarrow$ it has topo order



Algo:

Let L is empty list
while $G \neq \phi$
find a source v in G
Put v in the end of L
Delete v from G
Output L.



Time: $O(n^2) = \underbrace{O(n)}_{\text{iter}} \times \underbrace{O(n)}_{\text{cost to find source}}$.

Improve this $O(m+n)$.

Correctness: We can find a source v in G by Lemma 2a. and by G is DAG.

Let $v_1, v_2, \ldots, v_n$ be the order algo find.

For any edge (v_i, v_j)
we $i < j$ because the algo do job v_j only if all incoming edge to v_j is removed/done.

Proof of theorem follows from Lem 1 and Lem 2b.
