Discussion 3 Dec. 28. 2020

1. Short Answer

(a) If (u,v) is an edge in an undirected graph and during DFS, $post(v) < post(u)$, then u is an ancestor of v in the DFS tree.

True. Two cases: ~~pre~~ $[_u\ [_v\ ]_v\ ]_u$ and $[_v\ ]_v\ [_u\ ]_u$

the first case: u is an ancestor of v.

the second case (cross edge): v is already popped off while u hasn't been visited, this wouldn't happen because u is a neigh of v (when visiting v, u is explored).

Lemma: Undirected Graph only has tree edge and back edge. Forward edge is considered as a back edge.

(b) In a directed graph, if there is a path from u to v and $pre(u) < pre(v)$ then u is an ancestor of v in the DFS tree.

~~False. Consider a cycle~~.

False. Remember $\boxed{\text{that a path is not an edge}}$

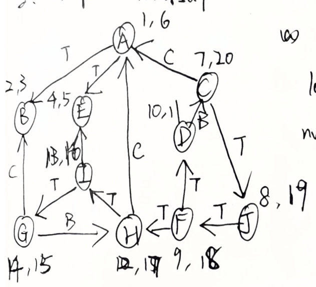
there is a path from u to v and $pre[u] < pre[v]$ but the tree is:

w
/ \
u v.

(c) In any connected undirected graph G there is a vertex whose removal leaves G connected.

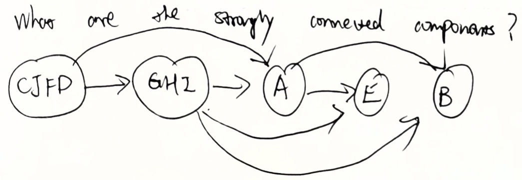
True. Considering DFS on graph. Any leaf in the DFS tree can be removed, leaving the graph still connected.

2. Graph Traversal



(a) perform DFS from A. label each node with pre and post number and mark the edge.

(b) What are the strongly connected components?



3. Finding Clusters.

Given a directed Graph $G = (V, E)$. $V = \{1 \ldots n\}$. For every vertex i compute $m(i)$ defined as follows:

$m(i)$ is the smallest j such from which you can r[each] vertex i.

(a) show that values $m(1) \ldots m(n)$ can be computed in $O(|V| + |E|)$ time.

Solution:

while exist unvisited nodes in G:
run DFS on G starting at an order of numbers (j)
for i visited do $m(i) = j$.

Reachable can be done by DFS and the smallest vertex reachable to i is completed by a sort of DFS procedure (by smaller first, larger last).

(b) Suppose $m(i)$ to be the smallest j can be reached from i. How to modify the answer.

Solution: $\boxed{\text{Reverse all edges on G}}$

4. BFS Intro

(a) Prove that BFS computes correct value of L_i. L_i is the set of vertices distance i from s,

By Induction,

for $i=0$, true

Assume $i=k$ true.

for $i=k+1$, every $k+1$ vertex ~~are~~ is adjacent to distance k, thus adds to $k+1$. No vertex more than $k+1$ can be added.

(b) Show that $T(n) = O(m+n)$

we check every vertex and edge overally.

(c) How to compute neighbored edge by BFS? (Integer)

Split the edge into couple of vertices.

(d) The run time for (c)? The bits taken? Is this a polynomial?

$T(n) = O(\sum_{e \in E} W_e)$ in G'. Bits: $O(\sum_{e \in E} \log W_e)$.

It takes exponential in the input size when W_e are large.