EL9343 Homework 4

All problem/exercise numbers are for the third edition of CLRS text book

---

1. Given an empty AVL tree and sequence {5, 1, 2, 4, 11, 17, 3, 7}, insert each element of the sequence to the AVL tree, one at a time, following the given order of the sequence. Draw all the 8 AVL trees after each insertion.

2. Given sequence { 2, 11, 5, 7 }, remove each element from the final AVL tree obtained in problem 1, one at a time, following the given order. Draw all 4 AVL trees after each deletion.

3. Exercise 22.2-6.

4. Given an O(V + E)-time algorithm to compute a path in a connected undirected graph G = (V, E) that traverses each edge in E exactly once in each direction.

5. Exercise 22.3-12.

6. Exercise 22.4-1.

7.Show how the procedure Strongly-Connected-Components works on the graph in Figure 1. Show the finishing times computed in line 1 and the forest produced in line 3. Assume DFS considers vertices in alphabetical order and and the adjacency lists are also alphabetical order.

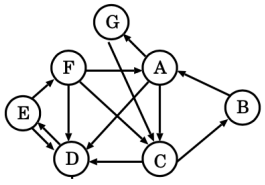


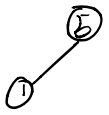
Figure 1: Directed Graph for Question 7

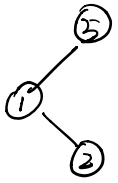
8. Problem 22-1.

9. Problem 22-3.

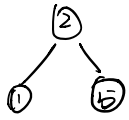
1. Iteration : (insertion)

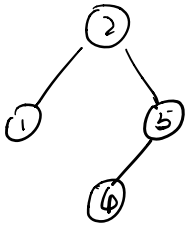
1. 

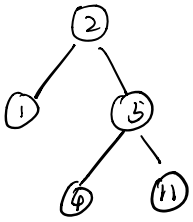
2. 

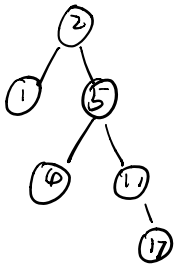
3. 

Double-Rotation →

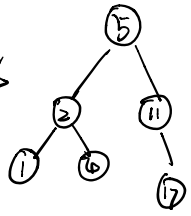


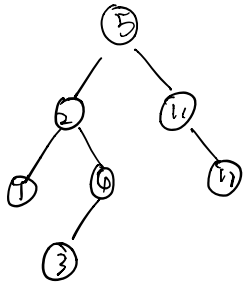
4. 

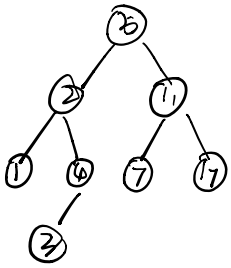
5 : 

6. 

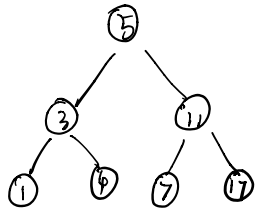
Single Rotation →

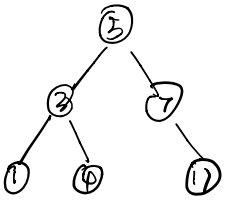


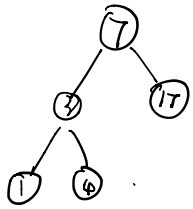
7. 

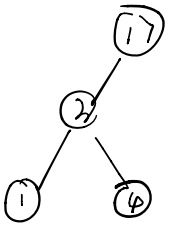
8 : 

2. Iteration : (Deletion)

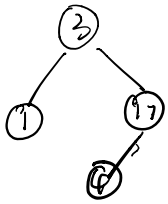
1. 

2. 

3. 

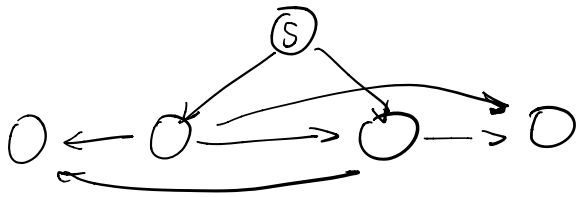
4. 

single - Rotation →

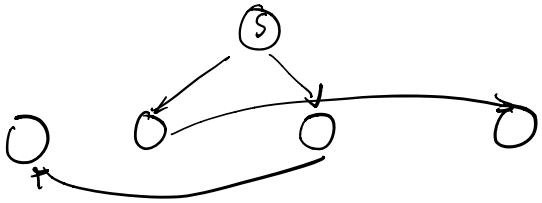


3. $G = (V, E)$, $s \in V$, and a set of tree edges $E_\pi \subseteq E$ s.t. $\forall v \in V$ the unique simple path in the graph (V, E_π) from $s \to v$ is the shortest path in G. yet the set of edges E_π cannot be produced by running BFS on G, no matter how the vertices are ordered in each adjacency list. (G, directed).

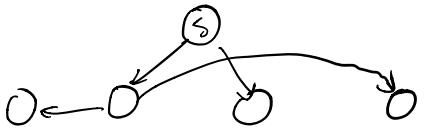
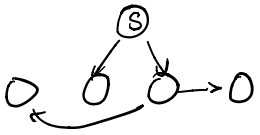
Sol:



$G = (V, E)$.



$G = (V, E_\pi)$

BFS Either produce.  or 

4. $O(V+E)$ algorithm that traverse each edge in E exactly one time in each direction on an undirected graph $G(V, E)$ that is connected.

Sol: For u in V:

DFS visit (u)

5. Show how to modify DFS s.t it assigns to each vertex v an integer label $v.cc$ between 1 and k, (no. of connected components of G.

Sol:

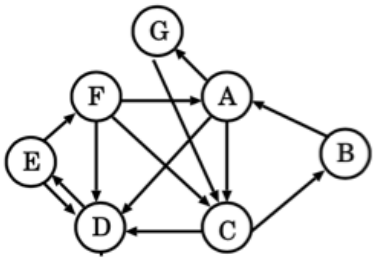
```
DFS(G) {
    for each u in V {
        color[u] = white;
        pred[u] = NULL;
    }
    k = 1;
    for each u in V
        if (color[u] == white) {
            k++;
            DFSvisit(u);
    }

DFSvisit[u] {
    color[u] = gray;
    u.cc = k;
    for each v in Adj(u) do
        if (color(v) == white) {
            pred[u] = v;
            DFSvisit(v);
        }
    color[u] = black;
}
```

6. Assumptions: For loops in DFS is in alphabetical order.

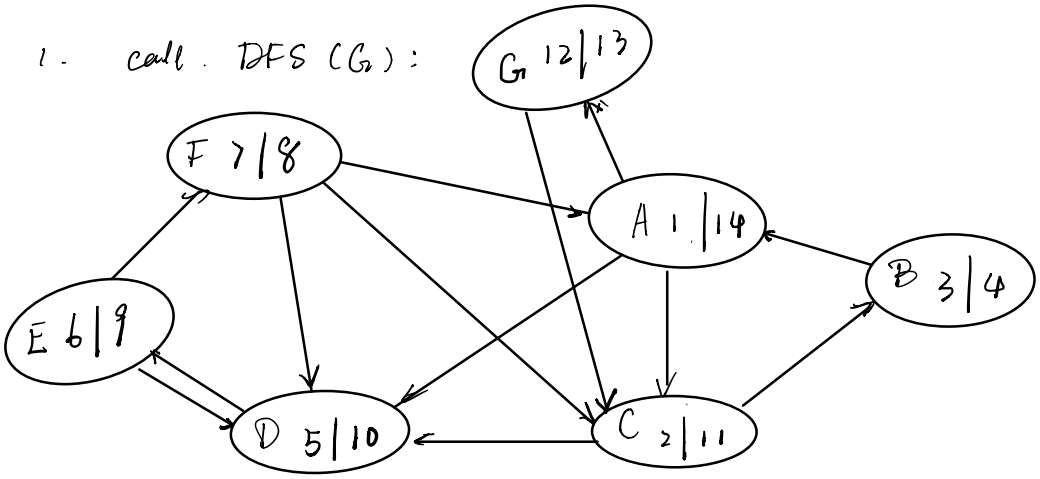
Sol: linked-list:

t, q, u, z, w, x, v, y, r, m, s, o, n, p

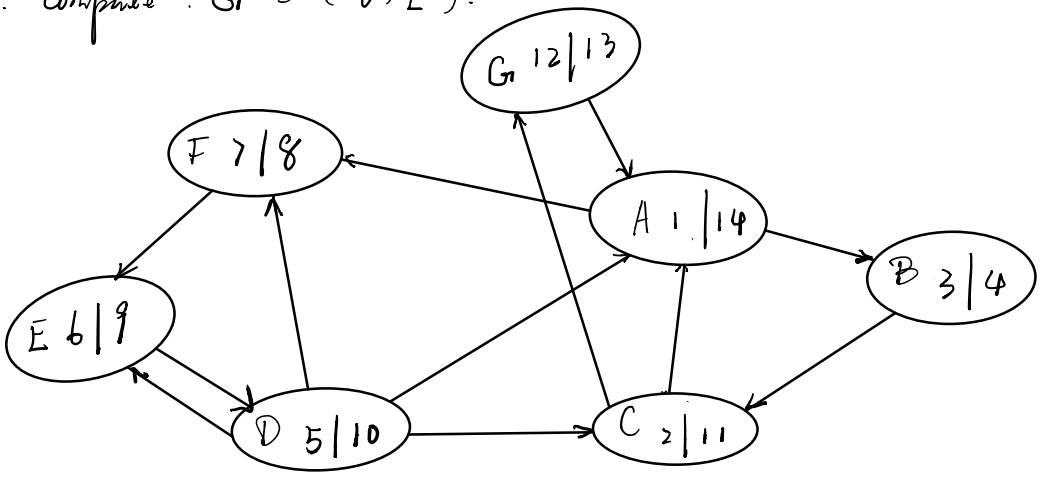
7.



Sol: 1. call DFS (G_r):

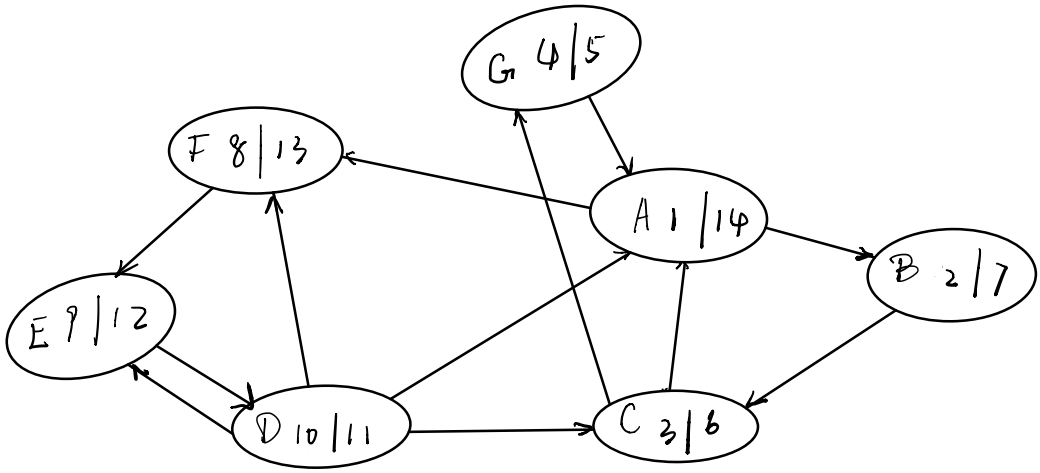


2. compute $G^T = (V, \bar{E})$.

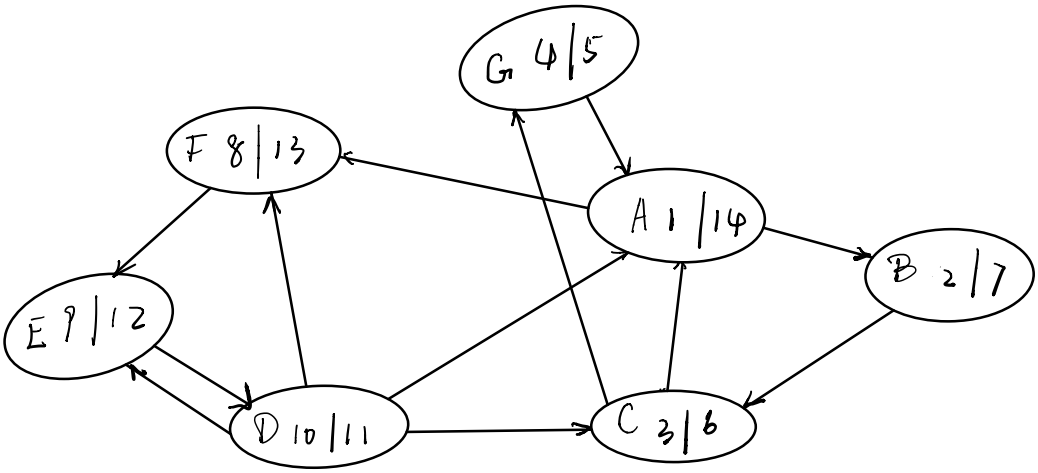


3. call $DFS(G^T)$ using the order of decreasing u.f.

which is: A, G, C, D, E, F, B



4. Output the tree computed previously.



which is exactly the same.

SCG $\{A, B, C, D, E, F, G\}$ since it's a tree computed from $DFS(G^T)$.

8.

Classifying Edges by BFS:

a. Prove: (For undirected graph)

1. No back edges and no forward edges.

Since it's an undirected graph. (u, v) where $v \in pred(u)$ does not exist because all the v's will be listed before encountering u; and if there are independent components, there will be no edges between them, otherwise not independent, therefore no forward edge exists.

2. $\forall$ tree edge (u, v), $v.d = u.d + 1$

Since $v.pred = u$, according to the algorithm.

$v.d = u.d + 1$.

3. $\forall$ cross edge (u, v), either $v.d = u.d$ or $v.d = u.d + 1$

Either u, v are listed together as the same vertex's descendant i.e. $v.d = u.d$. or u encounters v. $v.d = u.d + 1$.

b. prove: (directed graph).

1. No forward edges:

A forward edge (u, v) is when u has v as its descendant but never encounters v. which does not exist since BFS will list all u's descendant vertices.

2. $\forall$ tree edge (u, v), $v.d = u.d + 1$

the same argument holds as in (a)

3. $\forall$ cross edge (u, v), $v.d \leq u.d + 1$.

$\forall (u, v)$ that is a cross edge, v will be listed before u's descendant. therefore $v.d \leq u.d + 1$. by algorithm.

6. $\forall$ back edge (u,v). $0 \le v.d \le u.d$.

if (u,v) is a back edge, then v has been listed before u, therefore $v.d \le u.d$, and it's trivial that $v.d \ge 0$.

7. Euler-tour.

(An euler tour of a strongly connected, directed graph ($G=(V,E)$) is a cycle that traverses each edge of G exactly once, although it may visit a vertex more than once).

Show a) G has an euler tour if.f. in-degree(v) = out-degree(v) $\forall v \in V$.

proof: "$\Rightarrow$" in-degree(v) = out-degree(v) $\forall v \in V$

Define a walk $w: u \xrightarrow{*} v$ is closed is $u = v$. Let G be strongly-connected. let walk $w: v_0 \xrightarrow{l} v_l$, $v_0\, e_0\, v_1\, e_1 \cdots e_l\, v_l$ be the longest walk. since all the edges incident with v_l are used in w. (otherwise - we can extend the walk using some other edges unused) and v_l has even degree $d(v_l) = 2k$, and $v_l = v_0$. (otherwise if $v_l \neq v_0$ $d(v_l) = 2(k-1)+1$ supposing v_l occurs in w k times. the final edge ends in v_l but never comes out)

If w is not an Euler tour, there're some edge $\{e_w\}$ that are unused. let $e^* = (u,v) \in E$ be an unused edge where v is not traversed in w, there's a longer walk $w^*: v_0 \xrightarrow{*} u \to v \to u \xrightarrow{*} v_l$ thus constructing a contradiction. Therefore $w: v_0 \xrightarrow{*} v_0$ is the Euler tour of strongly connected graph G.

"<=" G is an Eulerian graph. let $u \xrightarrow{*} u$ be an Eulertour w

let $v \neq u, v \in G$ that occurs k times in w, then $d(v) = 2k$,

consider u., shift the tour by any number of vertices s.t. w can be

represented as : $v \xrightarrow{*} v$ $v \neq G$ and $v \in G$, $d(v) = 2k'$.

where k' is the times that the tour comes in or out through u.

and since G is strongly connected $\forall v \in G$ should satisfy the

same condition that the degree of v is even. which means that

the times the tour comes in and out are the same.

Q.E.D.