Discrete Mathematics HW10

20180617 You SeungWoo

November 25, 2023

Problem 1

Proof. Proof by induction. Let $P(j)$ be the given statement for j vertices. Consider $P(j=1)$. This means the tree contains only one vertex, unique. Therefore, $P(1)$ is true.
Suppose $P(j=n)$ is true, consider $P(j=n+1)$. Since the preorder and the number of children of each vertex are given, we can find one leaf vertex v which appears at first, because v has no children. Since v is in preorder, we can find its parent w(just preceding v).
Now, remove v in the given preorder and decrease the number of children of w by 1. This result is the same as the statement in $P(n)$. This means we get the unique tree except one vertex v. The last step is to return to origin: put v in its previous position and add 1 to the number of children of w. This last step is also unique(only one way), so we get the unique tree with $n+1$ vertices. Therefore, $P(n+1)$ is true.

□

Problem 2

Proof. Proof by (strong) induction. Let k be the number of operators. Then we need to prove the number of symbols is $k + 1$. Consider $k = 0$. Then by the definition of *Well-formed formula(WFF)*, it has the form x which is one symbol. Therefore, $k = 0$ is true.
Suppose $k \leq n$ is true, consider $k = n + 1$. Note that the form of WFF is $*XY$ if WFF has at least one binary operator. Remove the $*$ appears in front. Then the remains XY has n operators. Since X is WFF, it has $0 \leq m \leq n$ operators so has $m + 1$ symbols. Similarly, Y has $0 \leq n - m \leq n$ operators so has $n - m + 1$ symbols. Thus XY has n operators and $n + 2$ symbols. Now, add the removed operator. Then we get $n + 1$ operators and $n + 2$ symbols, true.

$\square$

Problem 3

Solution. Skipped. Very easy, so try it! The answer is:

- Prefix: $- \cup AB \cap A - BA$
- Postfix: $AB \cup ABA - \cap -$
- Infix: $(A \cup B) - (A \cap (B - A))$

$\square$

Problem 4

(a) *Proof.* Proof by induction. Note that we need to find *optimal vertex cover.* Denote the optimal vertex cover of a graph G is $V^*(G)$. Let $P(j)$ be the statement: $|V^*(K_j)| = j - 1$. Consider $P(j = 2)$. Then it is only a line, so just choose any one vertex. It is an optimal vertex cover, therefore $P(2)$ is true.
Suppose $P(j = n)$ is true. Consider $P(j = n+1)$. Remove one vertex(denote u) and its connected edges in K_{n+1}. Then the graph is K_n. By the induction hypothesis, it has $V^*(K_n)$ with size $n - 1$. This means K_n consists of:

- $n - 1$ vertices which are in $V^*(K_n)$
- 1 vertex(denote v) which is not in $V^*(K_n)$

Now, add u and its edges to origin. Then we get K_{n+1} with $n - 1$ vertices in $V^*(K_{n+1})$. If u is not in $V^*(K_{n+1})$, then there is the edge (u, v) which does not satisfy the definition of *vertex cover.* This implies $u \in V^*(K_{n+1})$. Therefore, $|V^*(K_{n+1})| = (n - 1) + 1 = n$, true.

$\square$

(b) *Proof.* Proof by induction. This problem is proved by the following:

> Show that $V^*(G) \leq n - 1$ for any simple undirected (connected) graph G.

Similarly in (a), let $P(j)$ be the statement: $|V^*(G_n)| \leq n - 1$. Note that G_n means any graph with n vertices. Consider $P(j = 2)$. Then it is only a line, so $V^*(G_2) = 1$. Therefore, $P(2)$ is true.
Suppose $P(j = n)$ is true. Consider $P(j = n + 1)$. Remove one vertex(denote u) and its connected edges in G_{n+1}. Then the graph is a subgraph of G_{n+1}(denote G_n). By the induction hypothesis, it has $V^*(G_n)$ with size at most $n - 1$. This means G_n consists of:

- at most $n - 1$ vertices which are in $V^*(G_n)$
- at least 1 vertex which are not in $V^*(G_n)$

Now, add u and its edges to origin. Then we get G_{n+1} with at most $n - 1$ vertices in $V^*(G_{n+1})$. Since there is at least 1 vertex which are not in $V^*(G_n)$, u must in $V^*(G_{n+1})$. Therefore, $|V^*(G_{n+1})| \leq (n - 1) + 1 = n$, true.

$\square$

Problem 5

Solution. Skipped. You can do it in 3 weighing.

□

Problem 6

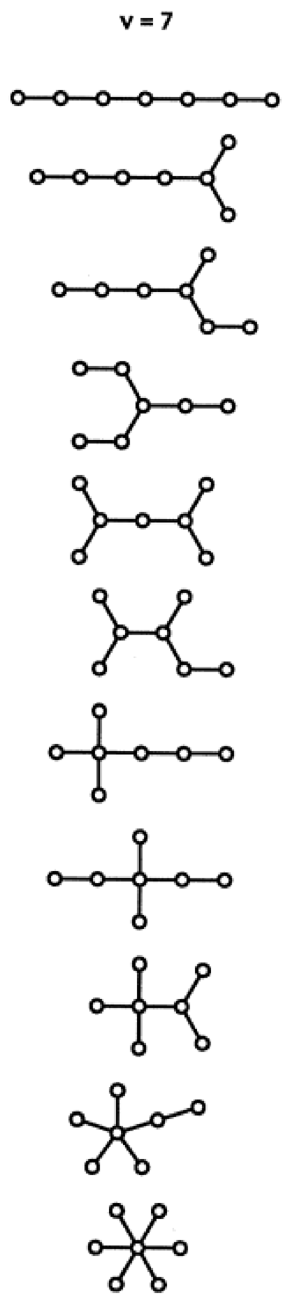
Solution. Skipped. The answer is

$$\begin{cases} 3 & \text{if only find bad coin} \\ 4 & \text{if find bad coin and determine lighter or heavier} \end{cases}$$

$\square$

Problem 7

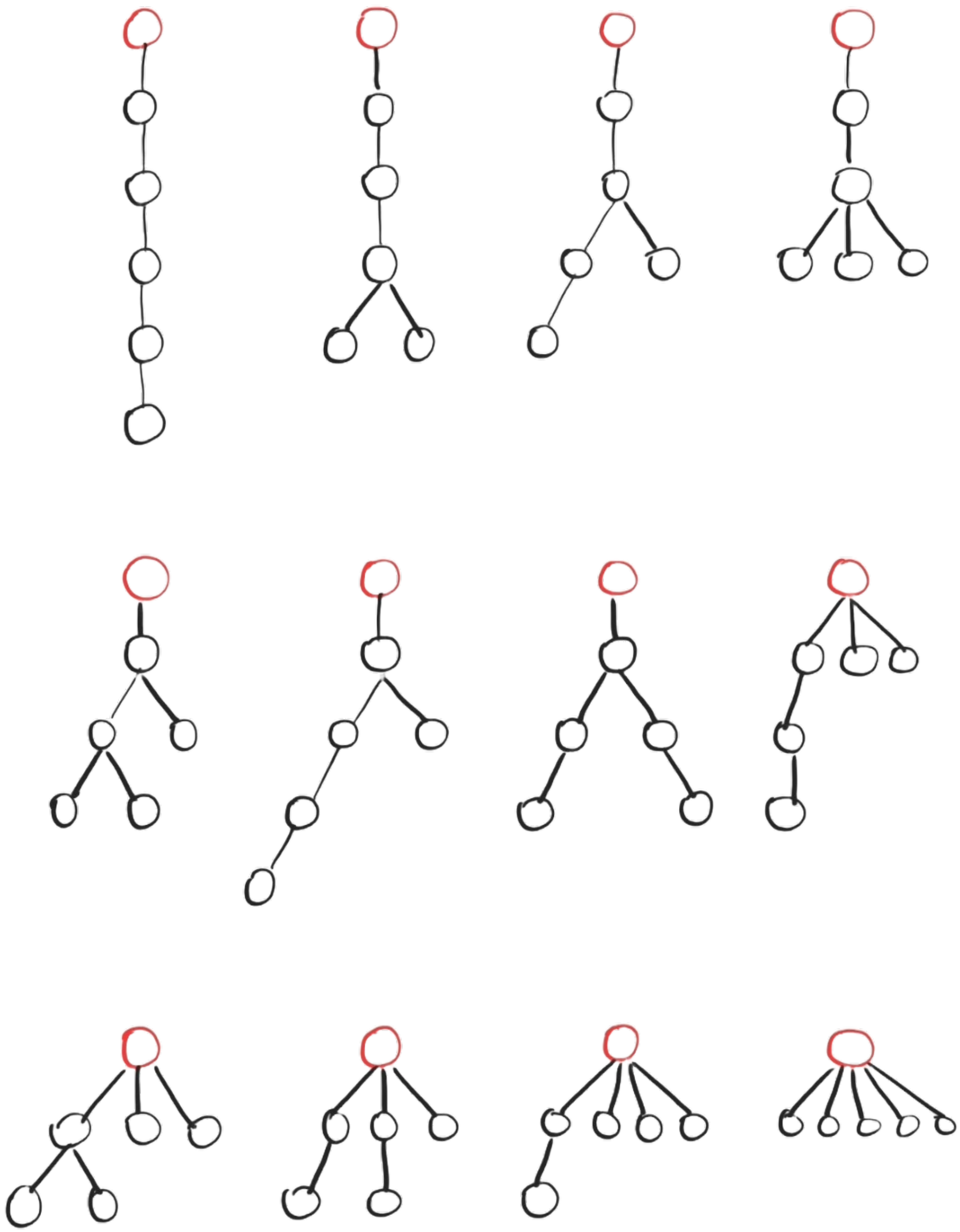
Solution. Total 11 trees.



□

Problem 8

Solution. I found 12, but I'm not sure if these are the only ones. Try it!



□

Problem 9

Solution. Follow the algorithm:

Step 1 Let $P = (a_1, a_2, \cdots, a_{n-2})$ be a Prüfer code. Then we have the tree with n nodes.

Step 2 Make a list $L = (1, 2, \cdots, n)$.

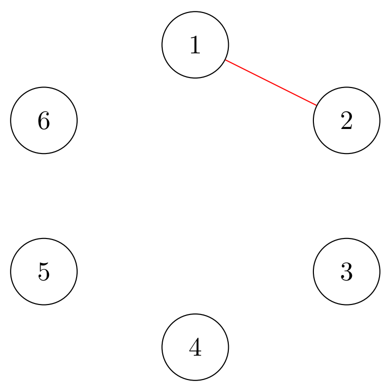
Step 3 Find the first number $a \in P$. Find the smallest number $b \in L$ but $b \notin P$. Remove them and add an edge (a, b) to the tree.

Step 4 If $P \neq \emptyset$, then return to the Step 3. If not, continue to next.

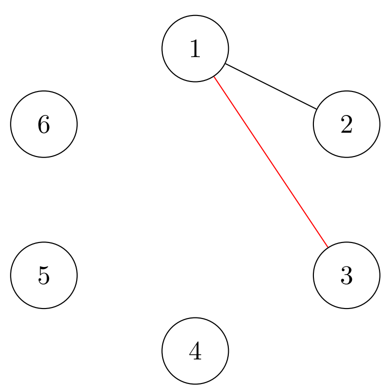
Step 5 We have only 2 values in L: let them be b_1 and b_2. Then just remove them and add an edge (b_1, b_2) to the tree. Done!

Note that the above algorithm gives well solution: the result is always given, well-formed, and unique.
Now, draw the first tree. Let $P = (1, 1, 1, 1)$. Then we have 6 nodes. Let $L = (1, 2, 3, 4, 5, 6)$.

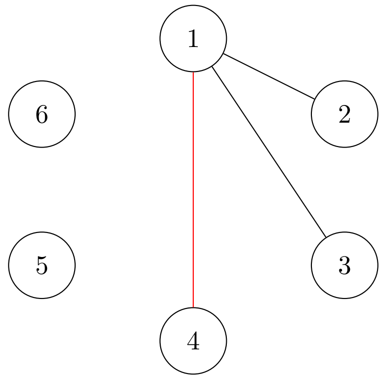
$$P = (\cancel{1}, 1, 1, 1)$$
$$L = (1, \cancel{2}, 3, 4, 5, 6)$$



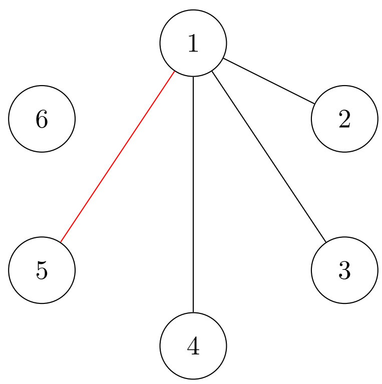
$$P = (\cancel{1}, \cancel{1}, 1, 1)$$
$$L = (1, \cancel{2}, \cancel{3}, 4, 5, 6)$$



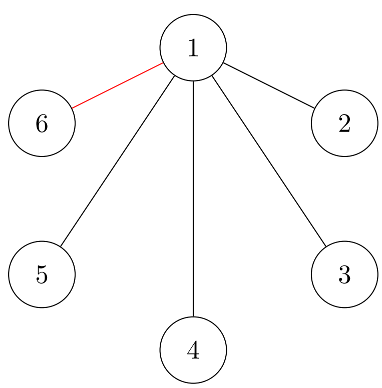
$$P = (\cancel{1}, \cancel{1}, \cancel{1}, 1)$$
$$L = (1, \cancel{2}, \cancel{3}, \cancel{4}, 5, 6)$$



$$P = (\cancel{1}, \cancel{1}, \cancel{1}, \cancel{1})$$
$$L = (1, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}, 6)$$

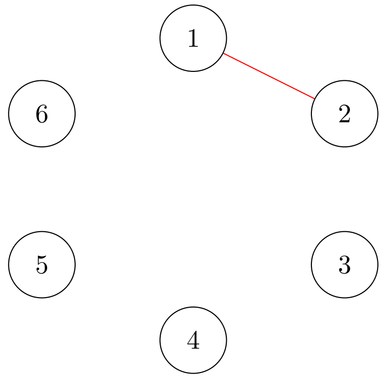


$$P = (\cancel{1}, \cancel{1}, \cancel{1}, \cancel{1})$$
$$L = (\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}, \cancel{6})$$

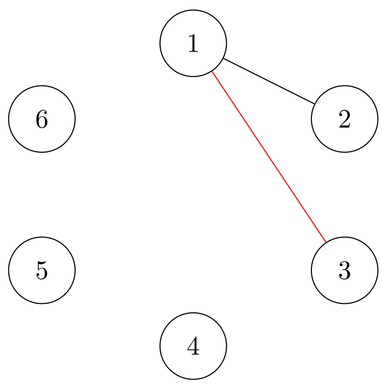


Draw the second tree. Let $P = (1, 1, 4, 4)$(You can make another code). Then we have 6 nodes. Let $L = (1, 2, 3, 4, 5, 6)$.

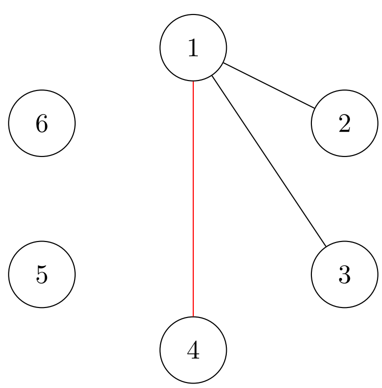
$$P = (\cancel{1}, 1, 4, 4)$$
$$L = (1, \cancel{2}, 3, 4, 5, 6)$$



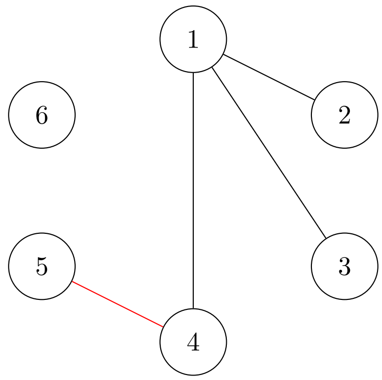
$$P = (\cancel{1}, \cancel{1}, 4, 4)$$
$$L = (1, \cancel{2}, \cancel{3}, 4, 5, 6)$$



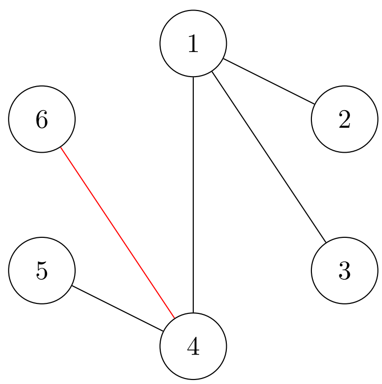
$$P = (\cancel{1}, \cancel{1}, \cancel{4}, 4)$$
$$L = (\cancel{1}, \cancel{2}, \cancel{3}, 4, 5, 6)$$



$$P = (\cancel{1}, \cancel{1}, \cancel{4}, \cancel{4})$$
$$L = (\cancel{1}, \cancel{2}, \cancel{3}, 4, \cancel{5}, 6)$$



$$P = (\cancel{1}, \cancel{1}, \cancel{4}, \cancel{4})$$
$$L = (\cancel{1}, \cancel{2}, \cancel{3}, \cancel{4}, \cancel{5}, \cancel{6})$$



$\square$

Problem 10

Solution. Follow the algorithm:

Step 1 Create adjacency matrix M for the given graph.

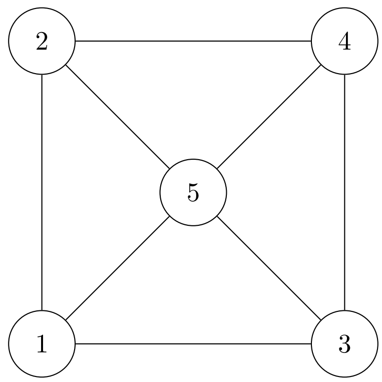
Step 2 Replace all the diagonals of M with the degree of nodes. For example, if $\deg(1) = 3$, then put $m_{1,1} = 3$ which is an element of M.

Step 3 Replace all non-diagonal 1's with -1.

Step 4 Remove any one row and one column of M. Let the remained be M^*.

Step 5 Calculate $\det M^*$. This is the total number of spanning trees. Done!

Note that Step 4 and Step 5 are the same as calculate co-factor for any element of M. Co-factor for all the elements will be same. Now, let the graph be



For the graph, the adjacency matrix is

$$M = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Replace all the diagonals of M with the degree of nodes.

$$M = \begin{pmatrix} 3 & 1 & 1 & 0 & 1 \\ 1 & 3 & 0 & 1 & 1 \\ 1 & 0 & 3 & 1 & 1 \\ 0 & 1 & 1 & 3 & 1 \\ 1 & 1 & 1 & 1 & 4 \end{pmatrix}$$

Replace all non-diagonal 1's with -1.

$$M = \begin{pmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 3 & 0 & -1 & -1 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}$$

Remove any one row and one column of M. I choose the last row and last column. Let the remained be M^*.

$$M^* = \begin{pmatrix} 3 & -1 & -1 & 0 \\ -1 & 3 & 0 & -1 \\ -1 & 0 & 3 & -1 \\ 0 & -1 & -1 & 3 \end{pmatrix}$$

$\det M^* = 45$. Therefore, there are total 45 spanning trees.
This algorithm comes from *Matrix Tree Theorem*, also called *Kirchhoff's Theorem*: [the number of nonidentical spanning trees of a graph G is equal to any cofactor of its Laplacian matrix.]

$\square$