MAT 145: PROBLEM SET 5

DUE TO FRIDAY MAR 1

Abstract. This problem set corresponds to the sixth week of the Combinatorics Course in the Winter Quarter 2019. It was posted online on Friday Feb 22 and is due Friday Mar 1 at the beginning of the class at 9:00am.

Information: These are the solutions for the Problem Set 5 corresponding to the Winter Quarter 2019 class of MAT 145 Combinatorics, with Prof. Casals and T.A. A. Aguirre. The Problems are written in black and the solutions in blue.

Problem 1. There are six different, i.e. mutually non-isomorphic, trees with six vertices. Draw these six trees.

Hint: For further practice, there is a unique tree with each of exactly 1, 2 or 3 vertices, there are two trees with 4 vertices and three trees with 5 vertices.

Solution. See Figure 1 for the six trees.

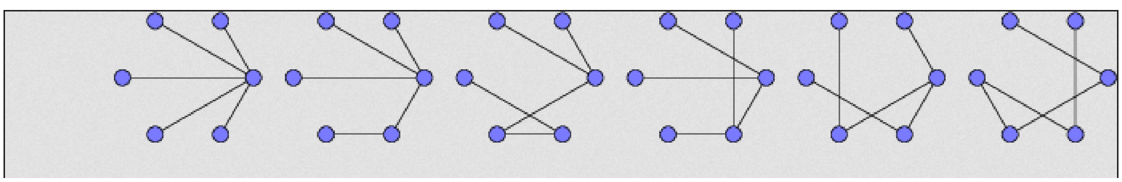


Figure 1. The six non-isomorphic trees on six vertices.

Problem 2. A *leaf* is a vertex of degree 1. Prove that if a tree $T = (V, E)$ has a vertex of degree d, then it has at least d leaves.

Solution. Let v denote such vertex. Suppose that $k \leq d$ of the neighbors of v are leaves. If $k = d$ we are done. If $k < d$ then there are $d - k$ vertices adjacent to v that have at least another neighbor. Again for each of those either they are a leaf or they have another adjacent vertex. Since T is a tree all the new vertices we get reiterating this procedure are distinct or else there would be a cycle. Eventually we reach at least d leaves. $\square$

Problem 3. (20 pts) Let $G = (V, E)$ be a connected graph, a connected subset $T \subseteq E$ is said to be a *spanning tree* for G if it satisfies the following two properties:

(i) Every vertex of G belong to an edge of T,
(ii) The edges in T form a tree.

Solve the following two problems.

(a) Find a spanning tree for each of the graphs in Figure 2. Are they unique in these cases ?

(b) Show that every connected graph has at least one spanning tree.

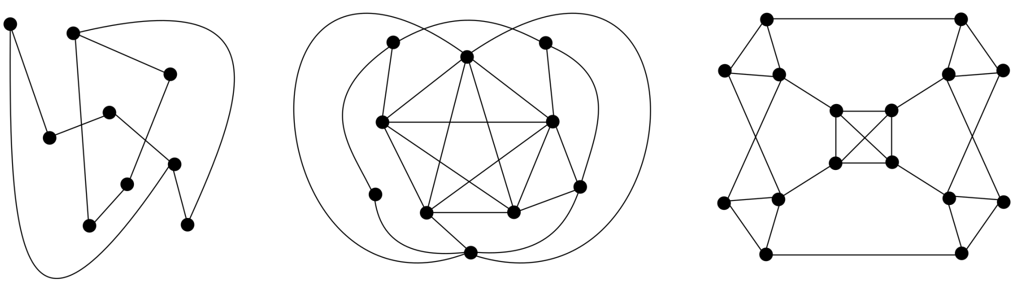


FIGURE 2. The three graphs for Part (a), find a spanning tree.

Solution.

(a) See Figure 3 for examples of spanning trees. Neither of them is unique since we can choose the edges we delete in order to break their cycles in several ways.

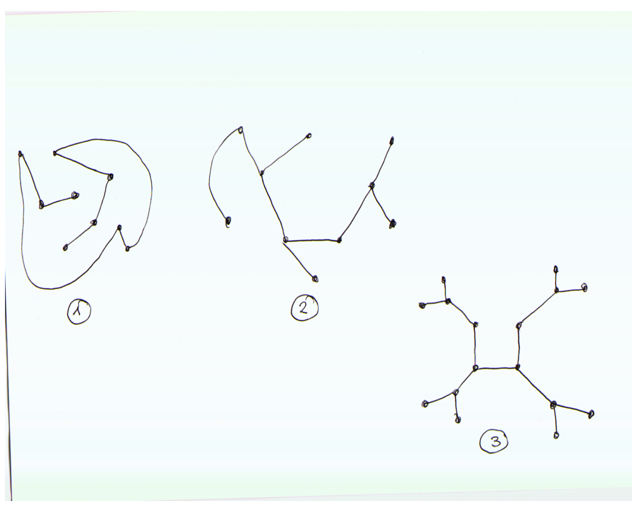


FIGURE 3. Spanning trees for the graphs in Figure 2.

(b) By definition, a connected graph is a tree if and only if it contains no cycles. Therefore, start with an arbitrary finite connected graph. We only have finitely many cycles. Every time we find one we remove one of the edges comprising it. This procedure keeps the graph connected and leaves the vertex set intact. After removing every cycle we are left with a spanning tree. $\square$

Problem 4. (20 pts) Let $n, m \in \mathbb{N}$ be two natural numbers.

(a) Show that the number of spanning trees of the complete graph K_n is n^{n-2}.

(b) (8.5.12) A (n, m)-dumbbell graph is constructed by considering the complete graph K_n on n vertices, the complete graph K_m on m nodes, and connecting them by a single edge. Figure 4 depicts the case $n = 4$ and $m = 5$. Find the number of spanning trees of a dumbbell graph.

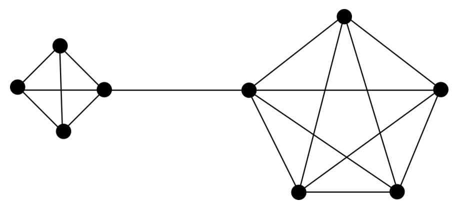


FIGURE 4. The $(4, 5)$-dumbbell graph.

Solution.

(a) The Prüfer correspondence gives a bijection between labeled trees on n nodes and sequences of integers $\{1, 2, 3...n\}$ of length $n-2$. Thus it suffices to establish a bijection between spanning trees of K_n and labeled trees. Note that any labeled tree on n nodes is automatically a spanning tree of K_n (it embdeds in it i.e we just add all the missing nodes to get K_n). So there are at least n^{n-2} spanning trees by the Prüfer correspondence. Conversely we label the nodes to K_n and create a spanning tree using the procedure in problem 3b. No matter how we remove edges are left with a labeled tree on n nodes so we have at most n^{n-2} distinct spanning trees. The two inequalities together imply that the number of spanning trees of K_n is n^{n-2}.

(b) Note that the (4,5)-dumbell graph is comprised by complete graphs on 4 and 5 vertices respectively joined by a bridge. Any spanning tree of the whole graph must use the bridge edge and will be a spanning tree within each of the 2 cliques with roots at the vertices of the bridge edge. However note since this is an undirected graph any vertex can be taken to be the root. By the formula from part (a) we have m^{m-2} spanning trees for the left clique and n^{n-2} for the right one. Thus the total number of spanning trees is $m^{m-2} \times n^{n-2}$. For the $(4, 5)$ dumbbell graph this would give us 2000 spanning trees.

Problem 5. (20 pts) The Prüfer correspondence[1] allows us to prove Cayley's Theorem. This problem gives direct practice on that correspondence.

(a) Draw the five labeled trees corresponding to the following five Prüfer codes:

$$\{4,4,4\}, \quad \{0,0,0\}, \quad \{1,2,4\}, \quad \{0,3,6,2,5\}, \quad \{2,3,4,5,6\}.$$

(b) Find the Prüfer code of the five labeled trees depicted in Figure 5.

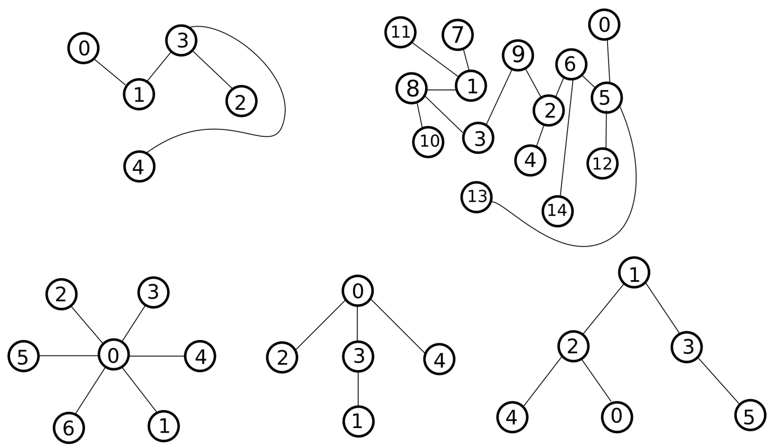


FIGURE 5. The five trees for Problem 5 Part (b).

Solution.

(a) See the Figure 6 below, where we have obtained these trees by using the following algorithm[2]: Given: a string S of length $n-2$ on alphabet $\{v_1, ..., v_n\}$, with $v_1 < v_2 ... < v_n$. Repeat until S is empty and alphabet has size 2:

(1) Identify the lowest letter in the alphabet that does not appear in thestring, v_i say, and the first element of the string,v_j say.

(2) Add v_i to the graph being constructed (if it isn't already there), and join it to v_j (adding v_j to the graph first if necessary).

(3) Remove v_i from the alphabet, and remove the first term from the string. Join the two remaining vertices in the alphabet.

(b) In clockwise order starting from the top left: $\{1,3,3\}$, $\{5,2,1,8,1,8,3,9,2,6,5,5,6\}$, $\{0,0,0,0,0\}$, $\{3,0,0\}$, $\{2,2,1,3\}$

[1]See Section 8.4 and the Lecture on Friday February 22nd.

[2]The convention that the node labeled with 0 is **always** the root means that if at any point in this algorithm the node 0 is a leaf with lowest degree we disregard it and take the next lowest leaf.

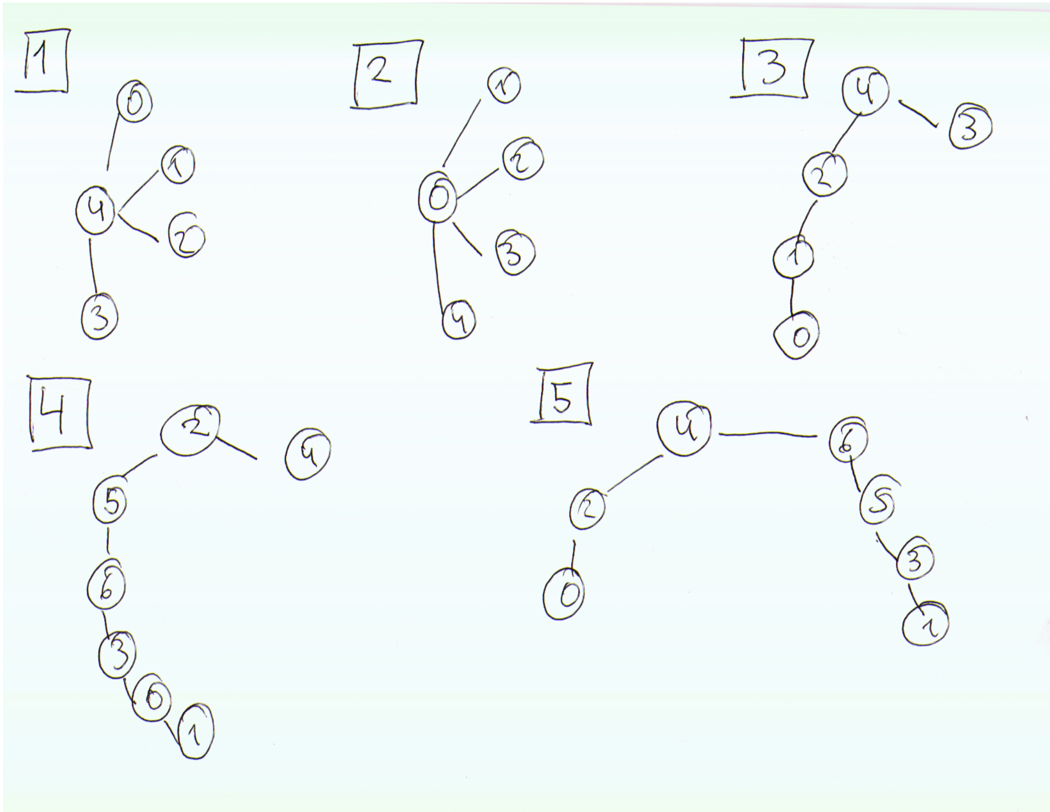


FIGURE 6. Solution to Problem 5.(a).

Problem 6. (20 pts) A *binary* tree is a rooted tree in which each vertex has at most two children, which are ordered, and are referred to as the *left child* and the *right child*. For instance, there are 2 binary trees in two vertices and 5 binary trees in three vertices, depicted in Figure 7. Let $n \in \mathbb{N}$ be a natural number.

(a) Show that the number of binary trees in n vertices is $C_n = \frac{1}{n+1}\binom{2n}{n}$.

Hint: Notice that the numbers $C_n = \frac{1}{n+1}\binom{2n}{n}$ *satisfy the recursion*

$$C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}.$$

(b) Establish a bijection between binary trees in n vertices and triangulations of regular convex $(n+2)$-gon. How many triangulations does an octagon have ?

Solution.

(a) To show the equality we wish to establish a bijection between pairs of binary trees on i and $n-i$ edges and binary trees on $n+1$ edges. Note by inductive assumption the right hand side is counting the number of binary trees rooted trees such that one of the descendants of the true is the binary tree on i vertices and the other one is a binary tree on $n-i$ vertices. Together with the root they form a (rooted) binary tree on $n+1$ vertices. Counting all the cases

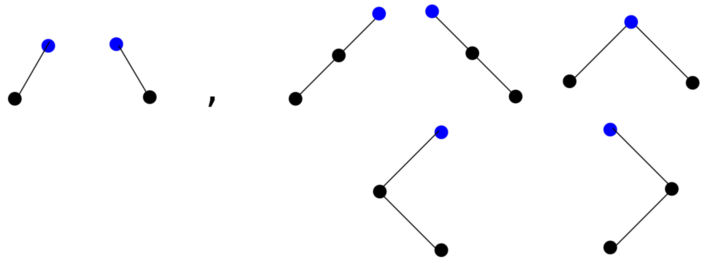
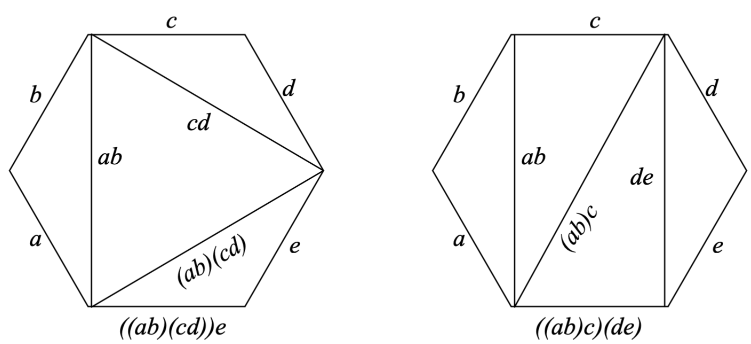


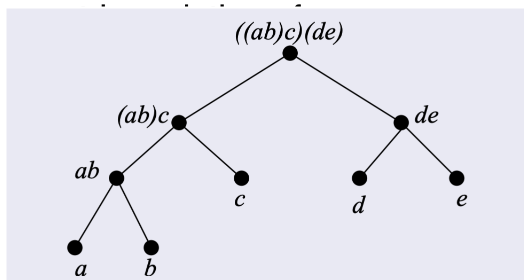
FIGURE 7. The two rooted binary trees with two vertices (Left) and the five rooted binary trees with three vertices (Right).

$0 \leq i \leq n$ we get $C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}$. Note that in the setting of the binary tree the labeling distinguishes between the two sides so we are not overcounting by summing $C_i C_{n-i}$ and then $C_{n-i} C_i$. Conversely, suppose we have a rooted tree on $n + 1$ vertices. Deleting the root we are left with binary subtrees with the left and right descendant as their roots. We can have $0 \leq i \leq n$ vertices on the left subtree (i.e possibly empty or possibly the whole rest of the tree and every possibility in between) and $n - i$ vertices on the right one. Adding all these possiblities we get the recursion $C_{n+1} = \sum_{i=0}^{n} C_i C_{n-i}$

(b) From Problem Set 2, we know that there will be exactly n with $n-1$ diagonals. The goal is to identify these the $2n + 1$ edges ($n + 2$ from the original n-gon and $n - 1$ from the diagonals) in this triangulation with the $2n + 1$ vertices in a binary tree of size n (note that it has by definition n internal nodes and $n + 1$ leaves). Use $n + 1$ letters to encode all but one of the sides of the n-gon (these will corresponds to the leaves of the binary tree). For each subsequent unlabeled edge in the triangulation let's denote it by the concatenation (with parentheses) of the two labels for the other edges of the triangle. Now there is clearly a bijection between words on $n + 1$ letters with closed parentheses and binary trees. Namely each letter is a leaf and each occurence of parenthesis is an internal node with its two descendants being the vertices encoded by the substrings within it.

Alternatively, for each triangulation we can take the polyhedral dual where each triangle becomes a node and nodes are adjacent if and only if their corresponding triangles are adjacent. The resulting graph will be a binary tree and different triangulations will induce distinct trees. See Figure 9 below.





(A) Binary tree corresponding to the left triangulation in the previous figure

FIGURE 8. Bijection between triangulations and binary trees.

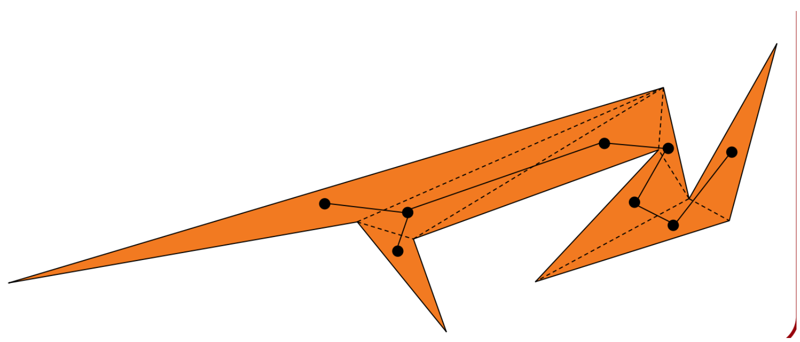


FIGURE 9. Solution to Problem 6.(b) via dual graph.

Problem 7. (20 pts) Solve the following two problems.

(a) Let $n_1, n_2, n_3 \ldots, n_d$ be a sequence of d natural numbers and $d \geq 2$. Show that there exists a tree $T = (V, E)$ with vertex degrees exactly $n_1, n_2, n_3 \ldots, n_d$ if and only if $n_1 + n_2 + n_3 + \ldots + n_d = 2d - 2$.

(b) Let $T = (V, E)$ be a tree with no vertices of degree 2. Show that there are more leaves[3] than non-leaves.

Solution.

(a) One direction is simple. In any graph we have that $\sum_{i \in V} n_i = 2|E|$ and in particular for trees $|E| = |V| - 1$ which shows the desired equality. For the converse suppose we proceed by induction to show that if a graph satisfies the condition

[3]By definition, as explained in Problem 2 above, a *leaf* is a vertex of degree 1.

on the degree sequence then it is a tree. IF $|V| = 1$ then $n_1 = 0 = 2|V| - 2$ and T is clearly a tree. For the inductive step suppose that for all $|V| = d \leq N$ the identity $n_1 + n_2 + n_3 + \ldots + n_d = 2d - 2$ implies the graph $T = (V, E)$ is a tree. Adding the $d + 1$-th vertex (denote this new graph by T') increases the right hand side by 2 and hence so must the left hand side (this equality is the assumption for this direction). On the left hand side however we are adding $n_{d+1} \geq 1$. Since each edge connects to vertices if $n_{d+1} > 1$ we will be increasing more than n_i by at least once hence resulting in the left hand side increasing by at least 4. Therefore $n_{d+1} = 1$ and some other n_i increases by exactly 1. Note that this precisely corresponds to $d + 1$ being a leaf attached to the old tree so we generated a new tree.

(b) Since T is a tree we know that $|E| = |V| - 1 = d - 1$. Then let n_l denote the number of leaves so that $d - n_l$ is the number of nonleaves. The leaves have clearly degree 1 and the nonleaves have degree at least 3 therefore by the degree formula:

$$2(d - 1) \geq n_l + 3(d - n_l) \implies n_l \geq \frac{d}{2} + 1$$

Thus leaves constitute more than half the nodes, as desired. □

Problem 8. Let $G = (V, E)$ be a connected graph. Show that G is a tree if and only if any three pairwise vertex-intersecting paths in G have a common vertex.

Solution. First let's assume G is a tree and P_1, P_2, P_3 are pairwise intersecting path. Let $P_1 = v_1 v_2 \ldots v_n$. By assumption there exist i, j such that $v_i \in P_2$ and $v_k \in P_3$. Without loss of generality let $i < k$ and assume by contradiction that there are no common vertices to all three paths. By this assumption then there exists $i \leq j < k$ such that $v_j \in P_2$ but $v_{j+1} \notin P_2$ then the edge $v_j v_{j+1}$ is in neither P_2 nor P_3. Now deleting this edge will turn the tree into a disconnected graph. Since P_2 and P_3 are paths each has to be within a connected component. However note that $v_i \in P_2$ and $v_k \in P_3$ are in different connected components thus contradicting the assumtption that P_2 and P_3 intersect.

Conversely, suppose G is not a tree then there exists a cycle and it will be of length at least 3. Let v_1, v_2, v_3 be vertices in this cycle and P_1 the path connecting v_1 and v_2, P_2 the path connecting v_2 and v_3 and P_3 the path connecting v_3 and v_1. Then these paths intersect pairwise but have no common intersection vertex. □