

PROBLEM SET 2 SOLUTIONS
MAS341: GRAPH THEORY

1. QUESTION 1

The triangular prism graph is a graph with 6 vertices and 9 edges, consisting of two triangles, with three more edges connecting corresponding vertices of the triangles.

As of this writing (hopefully one of you will fix it; I've never edited wikipedia), the Wikipedia article for prism graphs has a mistake in the formula for the number of spanning trees (if you follow the link to their source, they just miscopied something).

Prove that, in fact, the triangular prism graph has 75 spanning trees, not 78. (One possible way to organize the count: consider the three edges NOT contained in any triangles. There are 12 spanning trees that contain all three of these, 36 spanning trees that contain 2 of these edges, and 27 spanning trees that contain just one of these edges).

Proof. We follow the hint. Consider three edges connecting the two triangles; since removing these three edges disconnects the graph, our spanning tree must contain at least one of these edges. We consider three cases, depending on how many of these edges the spanning tree contains.

Consider the case when there is just one of these three edges; there are three possibilities for which edge it could be. Once we have done this, we need four more edges, and we need a spanning tree of each of the two triangles. Thus, we need to forget one edge from each triangle. There are three ways from each triangle to choose this edge, and thus after the initial choice of three edges, we have 9 choices for the rest of the spanning tree, and so there are

$$3^3 = 27$$

spanning trees that involve one of these edges.

Now, consider the case where the spanning tree contains two of three edges connecting the triangles. Then we are forgetting one edge, and so there are three ways to choose these two edges. After we've chosen these edges, we have three edges that will fill the rest of the spanning tree. We will need one side to contain two edges and one side to contain 1 – there are 2 ways to choose which side of the triangle has two edges.

The side that contains two edges will automatically connect the three vertices of that triangle, and so we can choose any of the 3 edges to forget.

On the side that only has one edge, there will be one vertex that does not yet have an edge connecting to it, so we will need to contain that vertex, and hence only have 2 ways to choose that tree. Thus, we see there are $3 \cdot 2 \cdot 3 \cdot 2 = 36$ spanning trees that contain 2 of long edges.

Finally, consider the case where our spanning tree contains all three of the spanning tree. There are two more edges our tree needs to contain, and we have two cases: where the two edges are in the same end of the prism, and where we have one edge from each end of the prism.

If both edges are from the same end, there are 2 ways to choose the end, and then 3 ways to choose what those three edges are; the resulting graph will automatically be a tree, and so we have 6 spanning trees of this type.

If we take one edge from each end, there are 3 ways to choose an edge from the first end. When we then go to choose an end from the other edge, we can't take the "same" edge, as this would result in a disconnected graph (and a loop). We have to take one of the remaining two edges. Thus, there are $3 \cdot 2 = 6$ spanning trees of this type, and hence 12 spanning trees total that contain all three of the long edges.

Putting it all together, we see there are 27 spanning trees containing 1 long edge, 36 spanning trees containing 2 long edges, and 12 spanning trees containing all three long edges, giving 75 spanning trees total.

□

2. QUESTION 2

2.1. **Part 1.** Prove that the number k appears m times in the Prufer code for a tree T if and only if the vertex labelled k in T has degree $m + 1$.

Show that this implies that the number of trees on n labeled vertices, where vertex i had degree d_i , is the multinomial coefficient

$$\binom{n-2}{d_1-1, d_2-1, d_3-1, \dots, d_n-1} = \frac{(n-2)!}{(d_1-1)!(d_2-1)! \cdots (d_n-1)!}$$

Proof. Recall that to construct the Prufer code, we first make a table containing 2 rows and $n - 1$ columns that records all $n - 1$ edges of the tree. The degree d_i of vertex i is the number of times i appears in this table.

The Prufer code is first $n - 2$ entries of the top row; thus, we see the proposition is equivalent to saying that in the remaining n entries of the table (the bottom row, and the last entry of the top row), every number appears exactly once. But we put a number in the bottom row when we delete it from the tree, so number can appear twice, and the final column of the table are the two vertices that are left after we've done this.

Put another way, conceptually, the number of times that i appears in the Prufer code is the number of time that vertex i is a parent of another vertex, but it also has

its own parent, thus the degree of vertex i is one more than the number of times i appears in the table.

Now, suppose we want to count the number of trees with n labeled vertices, where vertex i has degree d_i . Since vertex i has degree d_i , we have that i appears $d_i - 1$ times in the Prufer code. Thus, the Prufer code has length $n - 2$ times, and symbol i appears $d_i - 1$ times; this is exactly the multinomial coefficient

$$\binom{n-2}{d_1-1, d_2-1, d_3-1, \dots, d_n-1} = \frac{(n-2)!}{(d_1-1)!(d_2-1)! \cdots (d_n-1)!}$$

□