

PROBLEM SET 2 SOLUTIONS
MAS341: GRAPH THEORY

1. QUESTION 1

An edge e of a connected graph Γ is called a *bridge* if removing e from Γ results in a disconnected graph.

Prove that an edge e is a bridge if and only if it is contained in every spanning tree of Γ .

Proof. It is easiest to prove the contrapositives.

Suppose that e is not a bridge, that is, the graph $\Gamma \setminus e$ obtained by removing e from Γ is connected. Then in particular, $\Gamma \setminus e$ has a spanning tree T . But T is also a spanning tree of Γ that doesn't visit e . So if e isn't a bridge, there is some spanning tree that doesn't contain it.

Now, suppose that e is not contained in every spanning tree. Then there is some spanning tree T that doesn't contain it. But since T doesn't contain e , it is also a spanning tree of $\Gamma \setminus e$, and so $\Gamma \setminus e$ is connected.

Note that the two proofs are really the same in opposite direction: each step is in an if and only if: e is not a bridge if and only if $\Gamma \setminus e$ is connected if and only if $\Gamma \setminus e$ has a spanning tree if and only if Γ has a spanning tree that doesn't contain e if and only if e is not contained in all spanning trees of Γ .

□

2. QUESTION 2

2.1. **Part 1.** Recall the path graph P_n has n vertices v_i with v_i adjacent to $v_{i\pm 1}$. How many different labeled trees have the path graph P_n as their underlying graph?

Proof. Another way of phrasing this question is asking how many non-isomorphic labelings the path graph P_n has. It may help to look at an example, first – consider P_3 . It looks at first like it has 6 different labelings – 123, 132, 213, 231, 312, 321. But reading in the opposite order, we see that 123 is really the same labelled graph as 321, 132 is the same as 231, and 213 is the same as 312. So there are 3.

In general, since P_n has n vertices it looks at the beginning that there are $n!$ different labelings. However, P_n has an isomorphism group of size 2 – the identity isomorphism that sends every vertex to itself, and the isomorphism that reverses P_n and swaps vertices v_1 and v_n , vertices v_2 and v_{n-1} , and in general sends vertex v_i to vertex v_{n+1-i} . Thus, each nonisomorphic labelling of P_n will be obtained

twice in the $n!$ labeling, and there are $n!/2$ labeled trees that have P_n as the underlying unlabeled graph.

□

2.2. Part 2. From the Prüfer code of a labelled tree T , how can you tell if the underlying graph of T is isomorphic to P_n ? Count the number of codes that enumerate such labeled trees, and check that it agrees with your answer from Part 1.

Proof. The path graph P_n only has vertices of degree 1 and 2 – any other tree has a vertex of degree 3 or higher.

When we run the Prüfer code algorithm, we are constantly deleting leaves and recording the indices of their parents. If a number doesn't appear in the Prüfer code of T , then it was a leaf in T , and had degree 1. If a number appears once in the Prüfer code, then it is the "parent" of one vertex, and will be the child of another vertex, and so have degree 2. A vertex will have degree 3 or higher if and only if appears more than once in the Prüfer code. Thus, we see that the Prüfer codes that give labelings of P_n are precisely those in which no number is repeated more than once.

To count such codes, we need to count lists of $n-2$ numbers from 1 to n , with no number repeated. There are thus n choices for the first number, then $n-1$ choices for the second number, $n-2$ choices for the third number, down to $n+1-(n-2)=3$ choices for the last number. Thus, we have $n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 3$ such codes, which we see is equal to $n!/2$, the answer we got in Part 1.

□