Exercise Set Solutions #8

"Discrete Mathematics" (2025)

Exercise 8 is to be submitted on Moodle before 23:59 on April 14th, 2025

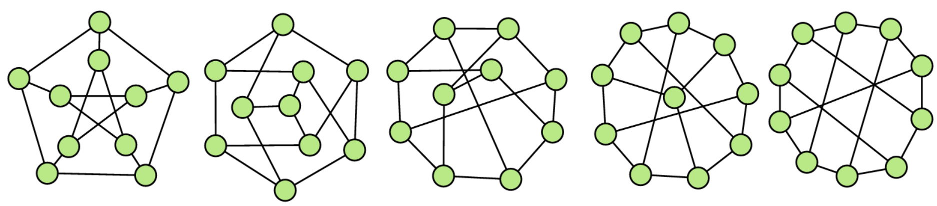
E1. **(a)** Is there a graph with 6 vertices with degrees $2, 3, 3, 3, 3, 3$?

> ***Solution:*** No. The sum of the degrees of all vertices in a graph is always even (see lecture).

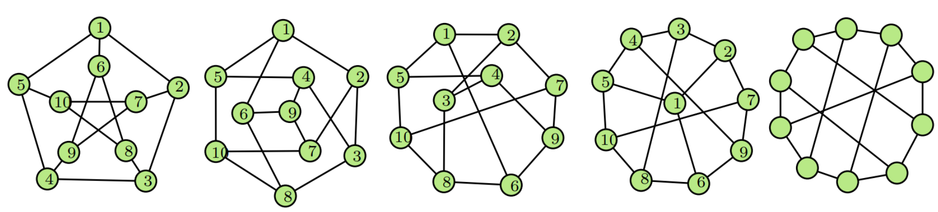
(b) How many labeled graphs are there with 4 vertices with degrees $1, 1, 2, 2$?

> The graph has to have the shape of a square with one side missing. Now, there are $\binom{4}{2} = 6$ possibilities to choose the leaves and for each choice of leaves there are two possibilities to order the vertices of degree two. In total that leaves us with 12 different graphs with 4 vertices.

E2. Are all the graphs below isomorphic to each other? Which ones are/are not? Why?
Hint: Try to move around the vertices of the graphs using a pencil and an eraser, or on a whiteboard.



> ***Solution:*** The following labelling reveals that the first four unlabelled graphs are isomorphic to each other. This graph is called a Petersen graph. The last graph contains a cycle of four edges, whereas there are no 4-cycles in the Petersen graph.



E3. Let $G = (V, E)$ be a graph and let $|V| = n$.

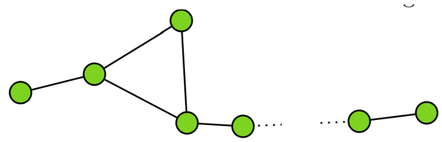
(a) What is the maximum number of automorphisms (isomorphisms from G to G) possible? Can you find a graph G with this number of automorphisms?

> ***Solution:*** It is impossible to have more than $n!$ automorphisms since each automorphism is a permutation of the n vertices. However, if the graph is the empty graph (when $E = \{\}$) or when it is the complete graph $E = \binom{V}{2}$ then each permutation is an automorphism and

we get that there are exactly n! of those

(b) What is the least number of automorphism possible? Can you find a graph with those many automorphisms?

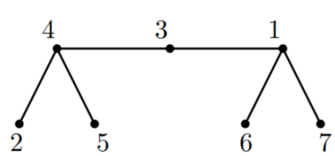
Solution: When $n \geq 6$, we can construct a graph with n vertices that has exactly 1 automorphism (the trivial one) to itself as shown below. For $n = 2, 3, 4, 5$ this construction will not work and we leave it to the reader to figure out what happens as it is not so difficult.

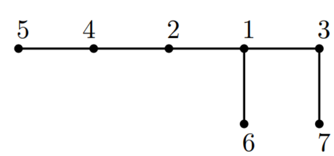


E4. Find the trees that have the following Prüfer sequences:

$$(4, 4, 3, 1, 1); (4, 2, 1, 1, 3)$$

Solution: The Prüfer sequence $(4, 4, 3, 1, 1)$ is of length 5 , therefore the corresponding tree has 7 edges.
(1) We start with the list $(1, 2, 3, 4, 5, 6, 7)$. The smallest number in the list which is not in the sequence is 2 . The first element in the sequence is 4 . Therefore we connect 4 with 2 by an edge and remove 2 in the list and 4 in the sequence.
(2) The smallest number in the list $(1, 3, 4, 5, 6, 7)$ which is not in the sequence $(4, 3, 1, 1)$ is 5 . The first element in the sequence is 4 . Therefore we connect 4 with 5 by an edge and remove 5 in the list and 4 in the sequence.
(3) The smallest number in the list $(1, 3, 4, 6, 7)$ which is not in the sequence $(3, 1, 1)$ is 4 . The first element in the sequence is 3 . Therefore we connect 4 with 3 by an edge and remove 4 in the list and 3 in the sequence.
(4) The smallest number in the list $(1, 3, 6, 7)$ which is not in the sequence $(1, 1)$ is 3 . The first element in the sequence is 1 . Therefore we connect 1 with 3 by an edge and remove 3 in the list and 1 in the sequence.
(5) The smallest number in the list $(1, 6, 7)$ which is not in the sequence (1) is 6 . The first element in the sequence is 1 . Therefore we connect 1 with 6 by an edge and remove 1 in the list and 6 in the sequence.
(6) The remaining list $(6, 7)$ has just two element. Therefore we connect these two vertices by an edge and stop.
We obtain the tree picture below on the left-hand side. For the sequence $(4, 2, 1, 1, 3)$ we obtain the tree on the right-hand side by using the same approach.





E5. **(a)** Describe which Prüfer codes correspond to stars (i.e. to trees where one vertex is connected to all other vertices).

Solution: A star on n vertices has $n-1$ leaves and the single vertex of degree $n-1$. So all the entries of the corresponding Prüfer code are the same. Such a Prüfer code we call constant. Conversely, if a Prüfer code with $n-2$ values is constant, then the tree has one vertex of degree $n-1$ and $n-1$ leaves (we construct the tree from the Prüfer code by starting with an edge and connecting $n-2$ vertices to the same endpoint of this edge). Thus stars exactly correspond to constant Prüfer codes.

(b) Describe what trees correspond to Prüfer codes containing exactly 2 different values.

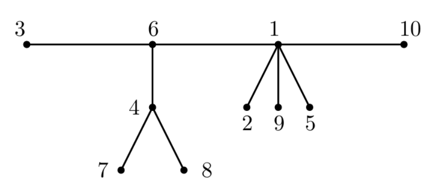
Solution: If the Prüfer code contains only two different values then the tree has exactly $n-2$ leaves and 2 non-leaves. Indeed, let us say without loss of generality that the two values appearing are 1 and 2 and let us say that 1 appears k times and 2 appears $n-2-k$ times. Then we know that we will have at least k vertices connected to 1 and at least $n-k-2$ vertices connected to 2 . When we arrive at the last number of the Prüfer code, if 1 and 2 were not already connected, we will connect them. When we connect them, if we had 1 written down in the Prüfer code this will increase the degree of 2 by one, namely, the degree of 2 will be $n-k-1$. Conversely, if 2 was written in the Prüfer code, then the degree of 1 will become $k+1$. Finally, we will have to add the edge whose column does not appear in Prüfer code, which will be $(n,1)$ if we had 1 written down before when we connected 1 and 2 , otherwise it will be $(n,2)$. Therefore, at the end we have $n-2$ leaves, one vertex of degree $n-k-1$ and one vertex of degree $k+1$ and these two vertices will be connected to each other.

Thus, the graph above corresponds to what we call a double-star: a graph with two vertices u, v connected by an edge, every other vertex is a leaf and connected to either u or v, at least one leaf is connected to each of u and v. Conversely, it is easy to see that double-stars have exactly 2 distinct values in their Prüfer codes. Thus, the codes described in this question exactly correspond to double-stars.

(c) And which trees have all distinct values in their Prüfer codes?

Solution: If all $n-2$ values in the Prüfer code are different, then there are only two vertices that do not appear in it. Thus, the corresponding tree has exactly two leaves. Furthermore, observe that all non-leaf degrees are 2 so the tree is a path. Conversely, a path on n vertices has exactly two leaves and $n-2$ distinct labels occur in its Prüfer code. The code has $n-2$ values so by the pidgin hole principle all values are distinct. Thus, these Prüfer codes correspond to paths.

E6. What is the Prüfer code of the following tree?



Solution: (1) The leave with the smallest number is 2 . Therefore we remove this vertex and put at first position the neighbor of 2 : (1,
(2) Next the smallest number of a leave is 3 , we remove it. The neigbor of 3 was 6 , so we get $(1,6,$
(3) Next the smallest number of a leave is 5 , we remove it. The neigbor of 5 was 1 , so we get

$(1, 6, 1$
(4) Next the smallest number of a leave is 7 , we remove it. The neigbor of 7 was 4 , so we get $(1, 6, 1, 4$
(5) Next the smallest number of a leave is 8 , we remove it. The neigbor of 8 was 4 , so we get $(1, 6, 1, 4, 4$
(6) Next the smallest number of a leave is 4 , we remove it. The neigbor of 4 was 6 , so we get $(1, 6, 1, 4, 4, 6$
(7) Next the smallest number of a leave is 6 , we remove it. The neigbor of 6 was 1 , so we get $(1, 6, 1, 4, 4, 6, 1,$
(8) Next the smallest number of a leave is 9 , we remove it. The neigbor of 9 was 1 , so we get $(1, 6, 1, 4, 4, 6, 1, 1$
Since the remaining tree has just 2 vertices, we are done. The Prüfer code is $(1, 6, 1, 4, 4, 6, 1, 1)$.

E7. Let v be a vertex of a labeled tree T. Suppose v has degree d, then show that the label of v appears in the Prüfer code of T exactly $d-1$ times.

Solution: Let us prove this by induction on the number of vertices n on T. It is clear that for the base case when $n = 2$, there is exactly on possible tree T with both the vertices being of degree 1 and the Prüfer code is the empty string.

For the general case, consider the algorithm that gives us a Prüfer code out of a tree T with n vertices. When it deletes the first leaf of the sequence, it gives us a tree T' of $n-1$ vertices. If this leaf was deleted out of v, then the degree of the v in T' is $d-1$ and the first entry of the Prüfer sequence is v. By induction therefore v will appear $d-2$ times in the remaining string. If the first leaf is not deleted from v, then v will have a degree of d in T' and it will appear $d-1$ times in next $n-3$ entries of the Prüfer sequence by induction.