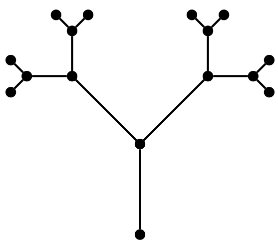
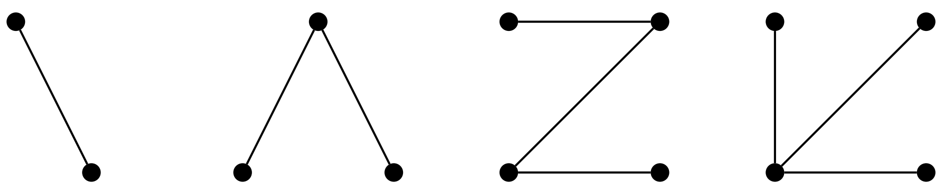
Class Four: Trees



A graph is **acyclic** if it contains no cycles, that is, if there are no walks from a vertex back to itself. We call an acyclic graph a **forrest** and refer to a connected, acyclic graph as a **tree**. A **leaf** is a vertex of degree one in a tree. We say that a tree is **irreducible** if it contains no vertices of degree 2. Below is an example of a forrest made up of four small trees two of which are irreducible.



Draw some small graphs and think about the following questions:

How many edges can a tree have?

What do the components of a forrest look like?

How many edges can a forrest have?

How big can the degree of a vertex in a tree be?

How small can the maximum degree of a vertex be in a tree?

Can all the vertices in a tree have odd degree?

If a tree has an even number of edges,
what can we say about the number of even vertices?

What is the average degree of a vertex in a tree?

What happens if you delete a vertex from a tree?

What happens if you delete an edge from a tree?

What happens if you add an edge to a tree?

How many cycles are created when you add an edge to a tree?

How many different paths can there be between two vertices in a tree?

How must a vertex attach to a tree so that the resulting graph is also a tree?

How many leaves can a tree have?

Draw all non-isomorphic trees with at most 6 vertices?
Draw all non-isomorphic trees with 7 vertices? (Hint: Answer is prime!)

Draw all non-isomorphic irreducible trees with 10 vertices?
(The Good Will Hunting hallway blackboard problem)

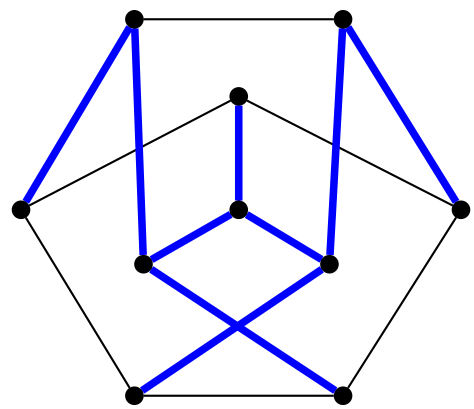
Lemma. *A forrest with n vertices and k components contains $n-k$ edges.*

Lemma. *A tree with at least two vertices must have at least two leaves. More generally, if a tree contains a vertex of degree Δ, then it has at least Δ leaves.*

Lemma. *The average degree of a vertex in a tree on n vertices is $2-\frac{2}{n}$.*

As we let the number of vertices grow things get crazy very quickly! This really is indicative of how much symmetry and finite geometry graphs encode. The sequence of number of non-isomorphic trees on n vertices for $n = 1, 2, 3, ...$ is as follows: $1, 1, 1, 2, 3, 6, 11, 23, 47, 106, 235, 551, 1301, 3159, ...$

A **subgraph** of a graph G is any graph obtained by deleting edges and vertices from G. A **spanning tree** of a graph G is subgraph of G obtained by



ONLY deleting edges from G which is itself a tree. For example in the below graph the blue edge give a spanning tree of the Petersen graph.

Draw some small graphs and think about the following questions:

If a graph has a spanning trees, need it be connected?

If a graph is connected, need it have a spanning tree?

How many spanning trees does a cycle have?

How many spanning trees does a tree have?

How many spanning trees does the wheel graph have?

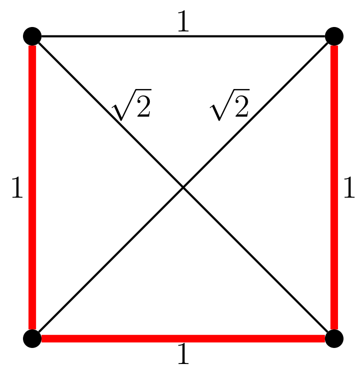
How many spanning trees does the fan graph have?

How many spanning trees does the complete graph have?

How many spanning trees does the Petersen graph have? (Very hard! There are 2000)

Lemma. *A graph is connected if and only if it has a spanning tree.*

Theorem (Cayley)**.** *There are n^{n-2} labelled trees on n vertices.*



A related algorithmic question arrises when considering graph with weights assigned to each edge. We call such a graph a **weighted graph**. Think of the vertices of a graph as corresponding to islands and the weights as corresponding to the cost of building a bridge between the two islands. Supposing that we only care about building bridges so that one can drive between every pair of islands and that we want to do this as cheap as possible. That is, given weighted edges of a graph we want to find a spanning tree such that the sum of the edges in the spanning tree are as small as possible. We call such a spanning tree a **minimum spanning tree**. For example, in the below weighted graph drawn is red is a minimum spanning tree.

Given a weighted graph,
how would you find a minimum weight spanning tree?

Lemma (Kruskal)**.** *Kruskal's greedy algorithm is an efficient procedures for finding a minimum weight spanning tree in a given weighted graph.*

While studying chemical compounds in the 1800s Cayley was the first to apply graph theory in the chemistry. We define the **bond** graph of a chemical compound as the graph who vertices correspond to atoms and whose edges correspond to various pairwise bonds. In particular, Cayley was trying to write down all isomers of the saturated hydrocarbons, that is all chemical compounds of the form C_kH_{2k+2}. In this situation hydrogen has valence 1, carbon valence 4 and there are no chemical chains (cycles), and so this question reduces to figuring out what all trees with vertices of degree only one or four look like.

Below are some small examples, some of which at the time of Cayley's work where know to exist and other whose existence he was able to predict.

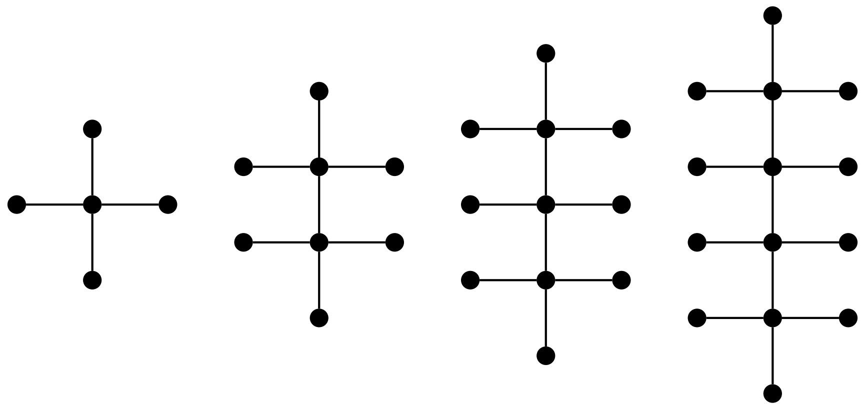


Figure 0.1: Bond graphs for Methane CH_4, Ethane C_2H_6, Propane C_3H_8 and Butane C_4H_{10}.

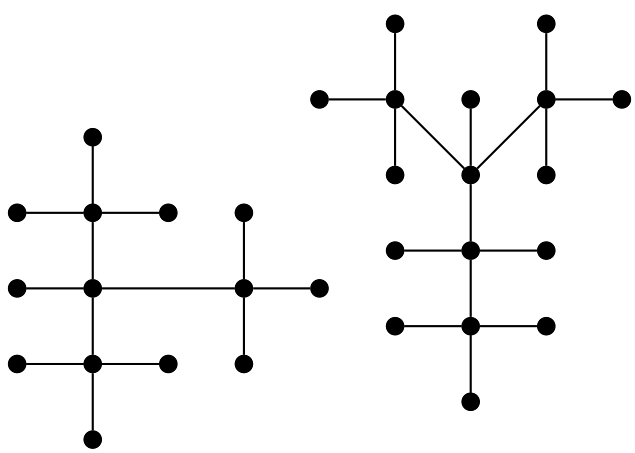


Figure 0.2: Bond graphs for Isobutane C_4H_{10} and Pentane C_5H_{12}.

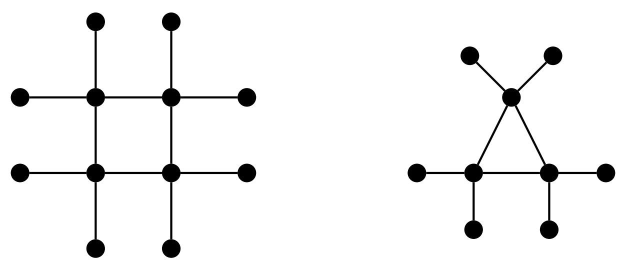


Figure 0.3: Bond graphs for Cyclobutane C_4H_8 and Cyclopropane C_3H_6.