# Spanning Trees and Decomposition



We give a more efficient way of counting the number of spanning trees in loopless graphs. As a second part, we are wondering if it is possible to decompose a graph into multiple copies of the same tree.

## Counting spanning trees, efficiently

Last lecture, we counted the number of spanning trees using the deletion-contraction process. That fell like a good process since it allowed us to count (for the first time) the number of spanning trees. However, the algorithm to do so has an exponential complexity (i.e. the number of steps required to make it work might be as big as (roughly) $2^{\#E}$).

The following theorem gives an efficient computation for the number of spanning trees.

### Theorem

Let G be a loopless graph and A be its adjacency matrix. Let L be the matrix with $l_{ij}=-a_{ij}$ and $l_{ii}=d(i)$, the degree of vertex i.

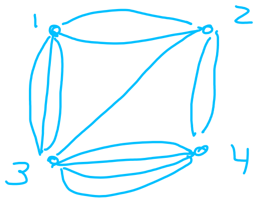
The number of spanning trees of G is any cofactor of L.

(Recall that the (i,j)-cofactor of the matrix M is computed by

$$(-1)^{i+j}\det(M_{ij})$$

where $M_{ij}$ is the matrix obtained from M by deleting its i-th row and j-th column.)

### Example




$$A=\begin{pmatrix}0&2&3&0\\2&0&1&2\\3&1&0&4\\0&2&4&0\end{pmatrix}$$

Just as we obtained last lecture, there are 106 spanning trees.

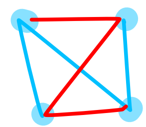
If you are interested in reading it, the proof can be found on pages 86-87 of the textbook.

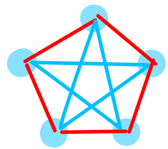
## Decomposition

Recall that a decomposition of a graph is a list of subgraphs in which every edge appears exactly once. This definition raises the following problem: When can we decompose a graph G into copies of H?

**Example**

Two copies of a self-complementary graph is a decomposition of a complete graph.





**Proposition**

If G decomposes into many copies of H, then

1) The number of edges in H divides the number of edges in G.

2) The maximum degree of H cannot be greater than the maximum degree of G.

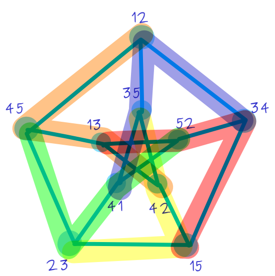
**Proof**

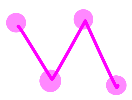
1) Assume there are m copies of H in G. Then, the number of edges in G is m times the number of edges in H.

2) Assume $\Delta(H) > \Delta(G)$. So there is a vertex v in H, and that vertex must appear in G as well. The copy in G may have more edges incident to it, but cannot have fewer. So v in G has degree $\Delta(H)$, contradicting the maximality of $\Delta(G)$.

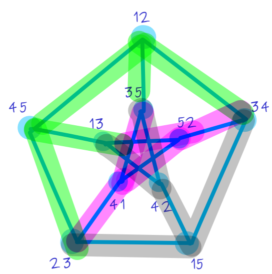
Are these two conditions sufficient for graphs to decompose into multiple copies of a graph? The following example will show this is not enough.

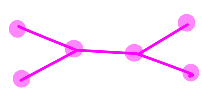
**Example:** Decomposition of the Petersen graph.



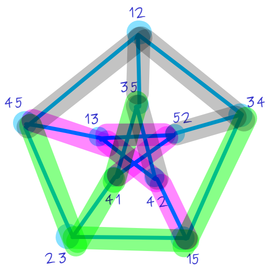



path

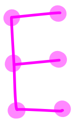


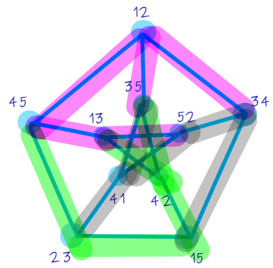



"H-graph"

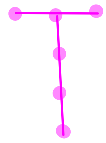



"E-graph"






"T-graph"



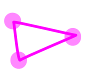
The Petersen graph decomposes into multiples copies of the following graphs:

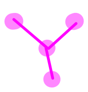
-
-
-
-

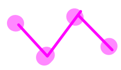
By the proposition above, 1, 3, 5 and 15 are the only possible numbers of edges that can appear in the smaller graphs. To prove the list is exhaustive, we must show that the H-graph, the E-graph and the T-graph are the only graphs with 5 edges that can occur in the decomposition, and the same has to be true for the path of length 3 compared to other graphs of size 3.

For the case of 5 edges:

To decompose it into 5 copies of a graph, with 3 edges, there are 3 options:







- The claw (star) is not possible because of the following proposition:

Proposition

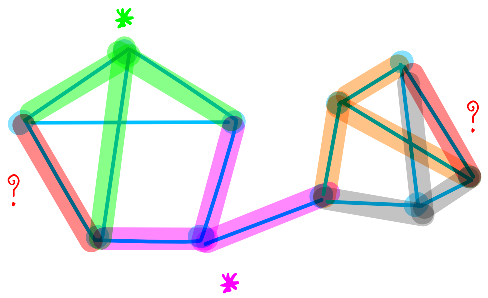
A k-regular graph can be decomposed into copies of stars with k edges if and only if the graph is bipartite.

Proof

<=

=> We prove the contrapositive: If a graph is not bipartite, it cannot be decomposed into stars with k edges. Assume it is not bipartite, so it contains at least one odd cycle. In this cycle, every other vertex must appear as the center of the star; otherwise, there are edges that cannot appear in the decomposition (like the ones in red below). Also, since every edge appears once, there cannot be two neighbouring vertices that appear, because every vertex takes all k the incident edges. If two adjacent vertices appeared, there would be an edge counted twice. So there cannot be an odd cycle in the graph to decompose it into stars.

only one edge left in cycle (around the pentagon)




This proposition allows us to conclude with the only possible decompositions for the Petersen graph.

Graceful labelings

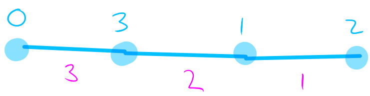
In general, the problem of decomposing a graph into many copies of graphs is a very hard one. Even the easier problem of decomposing it into trees is hard, as shown by the following conjectures:

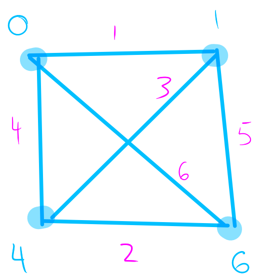
**Conjecture** (Ringel, 1964)
If $T$ is a fixed tree with $m$ edges, then $K_{2m+1}$ decomposes into $2m+1$ copies of $T$.

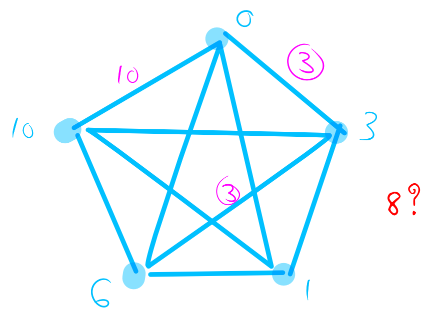
Note that despite multiple attempts to prove this conjecture, it is still open. Most attempts to solving it focus on graceful trees.

**Definition**
A graceful labeling of a graph $G$ with $m$ edges is a labeling of a graph with the numbers $\{0,\dots,m\}$ such that distinct vertices receive distinct labels and edges receive the difference of labels; the labeling is graceful if all the numbers $\{1,\dots,m\}$ appear on the edges of the graph.
A graph is graceful if it has a graceful labeling.

**Remark**
To make it possible to define such a labeling, a graph must have at least as many edges as the number of vertices minus one (which is the case of connected graphs, for example).



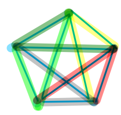

No graceful labeling

**Example**
All stars and paths are graceful.    Exercise: Find a proof!

**Conjecture** (Graceful Tree Conjecture – Kotzig, Ringel, 1964)
Every tree has a graceful labeling.

**Theorem** (Rosa, 1967)
If a tree $T$ with $m$ edges has a graceful labeling, then $K_{2m+1}$ has a decomposition into $2m+1$ copies of $T$.

Not an iff statement



Reference: Douglas B. West. Introduction to graph theory, 2nd edition, 2001. Section 2.2