Mathematical Foundations
of
Computer Science

CS 499, Shanghai Jiaotong University, Dominik Scheder

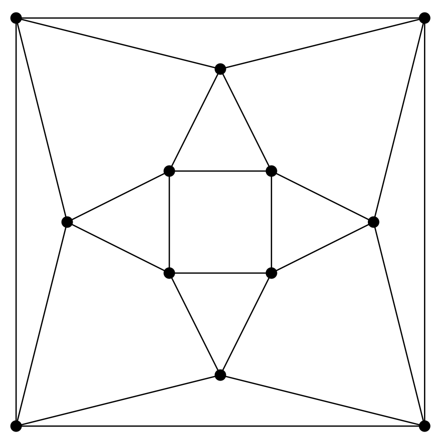
9 Hamilton Cycles, Hamilton Paths, and Non-isomorphic Trees

- Homework assignment published on Monday 2018-04-23
- Submit your first solution by Sunday, 2018-04-29, 18:00, by email
- Submit your final solution by Sunday, 2018-05-06.

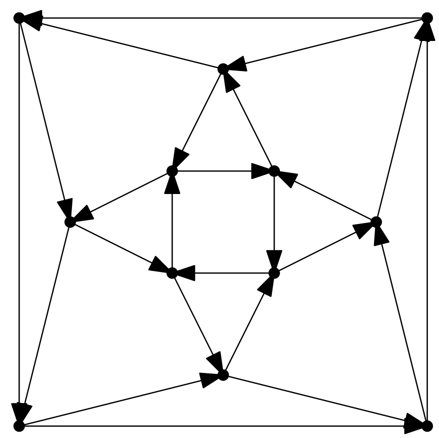
9.1 Regular Orientations of a Regular Graph

We call a graph d*-regular* if every vertex has degree d. A directed graph is (d, d)-regular if every vertex has d incoming and d outgoing edges.

Exercise 9.1. Show that in every 4-regular graph, you can orient the edges such that every vertex has two incoming and two outgoing edges, i.e., such that the resulting digraph is $(2, 2)$-regular. See the picture below for an illustration.



a 4-regular graph



a (2,2)-regular orientation

Proof. As the definition of Eulerian Walk, every edge is passed once and only once. If we walk into a vertice, we are sure to find another edge that has not been passed to leave the vertice. So for a vertice of degree 4, the two of them are in-degree and the other 2 are out-degree. □

9.2 Hamilton Cycles and Ore's Theorem

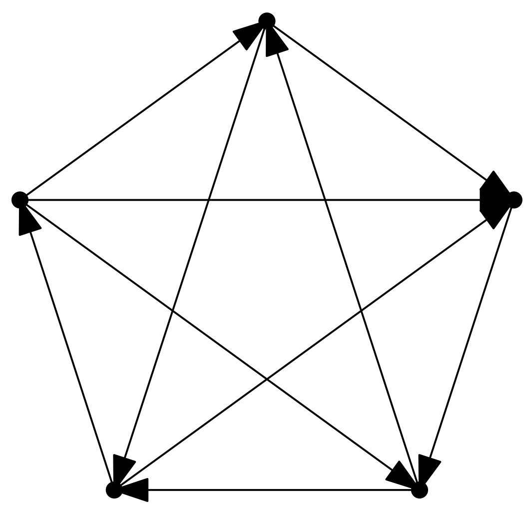
Consider K_n, the complete graph on n vertices. For $n \geq 3$, this obviously has a Hamilton cycle. How many edges do you have to delete from K_n to destroy all Hamilton cycles? That is, what is the smallest set S such that $\left(V, \binom{V}{2} \setminus S\right)$ has no Hamilton cycle? Let s_n denote the size of this set (this depends on n, thus the notation s_n). For example, $s_2 = 0$ since K_2 has no Hamilton path to begin with; $s_3 = 1$ since removing one edge from K_3 results in a graph without a Hamilton cycle.

Exercise 9.2. Find a closed formula for s_n and prove it! **Hint.** One part will be easy. For the other part, use Ore's Theorem.

Proof. $s_n = n - 2$
First, we prove that $s_n \leq n - 2$.
We can choose a vertice and delete $n-2$ edges of it, leaving only 1 edge. Since any vertice in a Hamilton cycle must have at least two edges, it's obvious there is no Hamilton cycle in this graph.
Second, we prove that $s_n \geq n - 2$.

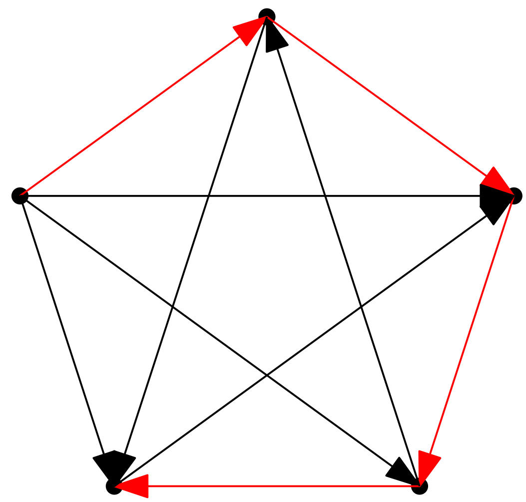
Assume we have a graph of n vertices with degree sequence $\{a_1, \ldots, a_n\}$. Notice that all a_i equals $n-1$. We delete k edges from it, $k < n-2$, and p vertices are involved. Without loss of generality, we assume that we get a new degree sequence $\{a_1 - k_1, \ldots, a_p - k_p, a_{p+1}, \ldots, a_n\}$, and $\sum_{i=1}^{p} = 2k$. For this new graph, we can find that the least possible sum of any two vertices' degrees is $a_i - k_i + a_j - k_j = 2n - 2 - (k_i + k_j)$. Since $k_i + k_j < k + 1 = n - 1$, the sum is bigger than $n-1$. According to the Ore's theorem, there must be a Hamilton cycle in this changed graph. So we need to delete at least $n-2$ edges eliminate Hamilton cycle from the graph.
Therefore, $s_n = n - 2$. $\square$

A *tournament* is a directed graph in which, for each pair $u, v \in V$, exactly one of the directed edges (u, v) and (v, u) is in the graph. Imagine a sports tournament in which every participant plays against every other exactly once. Draw an arc from u to v if u beat v in this tournament.



A tournament on five vertices.

Exercise 9.3. Show that every tournament has a *directed Hamilton path*, i.e., a sequence $u_1, u_2, \ldots, u_n$ such that $(u_i, u_{i+1}) \in E$ for all $i = 1, \ldots, n-1$. See the picture below.



The same tournament wiht a Hamilton path.

You probably won't be able to use the proof of Ore's Theorem directly, but you can use the proof idea.

Answer We can set up an algorithm.

First, we need to find a node and add it to the path, then colored the arrow both from the node and to the node another color(for example, blue).

Then, we find another node named x in the graph, observe the arrow with blue color connect with. There are three situations:

The first situation is that all the blue arrow connected with x point to x. Add x to the path and let x be the start node of the path.

The second situation is that all the blue arrow connected with x from x and point to other nodes. Add x to the path and let x be the tail node of the path.

The third aituation is that there exist the blue arrow not only point to x but also point to other nodes from x. Then we can find two nodes named m and n, m has a blue arrow point to x, x has a blue arrow point to n. Add x to the middle of m and n.

Implement this algorithm circularly and finally we can find Hamiton path.

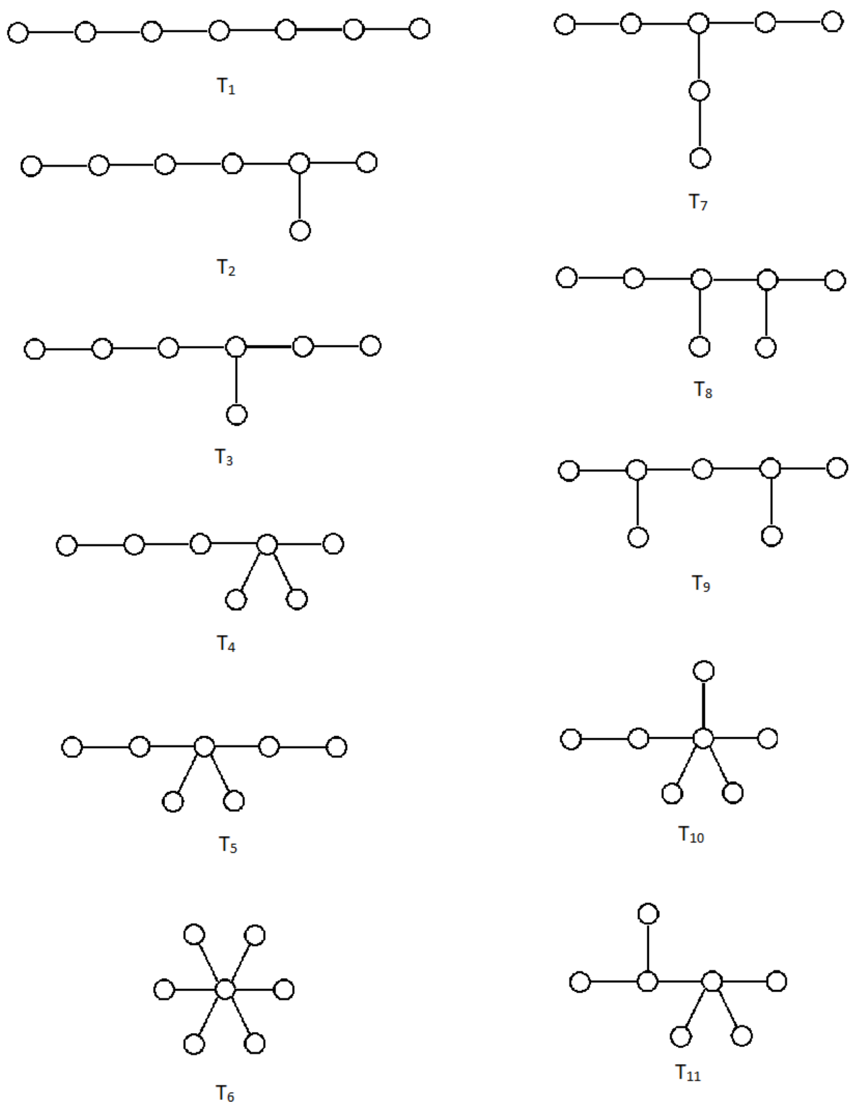
9.3 Isomorphism Classes of Trees

In the lecture (and in the videos) we have seen that the number of trees on vertex set $V = \{1, 2, \ldots, n\}$ is n^{n-2}. This however ignores isomorphisms. For example, there are $3^{3-2} = 3$ trees on vertex set $\{1, 2, 3\}$, but all those trees look alike (are isomorphic). On $\{1, 2, 3, 4\}$, there are 16 trees, but there are only two isomorphism classes: the path and the star. For five vertices, there are 125 trees but only three isomorphism classes: the path, the star, and the "T-shape" (see video on counting the number of trees). For $n = 6$ we get the path, the Y-shape, the Euro symbol, the Star Wars fighter, the Scandinavian cross, and the star, so six isomorphism classes (but a total of 1296 trees).

Exercise 9.4. List of isomorphism classes on seven vertices. That is, draw trees $T_1, \ldots, T_m$ on seven vertices such that no two of them are isomorphic but every tree on seven vertices is isomorphic to one of them. How many do you get?

n	1	2	3	4	5	6	7
number of isomorphism classes	1	1	1	2	3	6	?

Answer If $n = 7$, the number of isomorphism classes is 12, we draw the trees below.



Alright, so let's denote by t_n the number of isomorphism classes of trees on n vertices. That is, t_n is the largest number m such that we can find trees $T_1, \ldots, T_m$ on n vertices such that no two of them are isomorphic. We would like to have an exact and explicit formula for t_n, but that is probably too much to ask for. Instead, let us try to understand t_n approximately and asymptotically.

Exercise 9.5. Show that $t_n \leq 4^n$. Hint: Consider the video on the isomorphism problem on trees. It defines a way to encode a tree as a 0/1-sequence.

Proof. For a arbitrary ordered tree(an rooted tree, and each child of a vertex has an order), we traversal the tree in order from the root. For the first time we visit a vertex, we write 0, and when we leave a vertex and its subtree, we write 1. In this way, we can encode the tree with a $2n$-bit binary code, which is at most 4^n cases. Since two different ordered tree might be isomorphic(change the order of two children, for example), but obviously two trees that are not isomorphic can't be the same ordered tree, $T_n \leq 4^n$. $\square$

Exercise 9.6. Show that $t_n \geq \frac{e^n}{\text{poly}(n)}$, where $\text{poly}(n)$ is some polynomial in n. Hint: There are n^{n-2} trees on $V = [n]$. We group them together in "buckets" of isomorphic trees. How large can a bucket be? Answer this and then use Stirling's approximation for $n!$.

Proof. There are at most $n!$ isomorphisms of a given tree because there're $n!$ bijections on $[n]$, so the maximum size of "bucket" is $n!$. With Stirling's approximation, we can get a lower bound:

$$\begin{aligned} t_n &\geq \frac{n^{n-2}}{n!} \\ &\approx \frac{n^{n-2}}{\sqrt{2\pi n}(\frac{n}{e})^n} \\ &= \frac{e^n}{\sqrt{2\pi} n^{2.5}} \end{aligned}$$

$\square$

****Exercise 9.7.** *Try to improve those bounds. That is, find some $a < 4$ such that $t_n \in O(a^n)$ or some $b > e$ such that $t_n \in \Omega(b^n)$. Any improvement will be kind of interesting. Aim for simple proofs!*

Remark. The "true" rate of growth is known by a result of George Pólya but apparently it is quite difficult (I write "apparently" because I have never studied this work).

We have no idea yet.