18.212: Algebraic Combinatorics

Andrew Lin

Spring 2019

This class is being taught by **Professor Postnikov**.

April 29, 2019

Today we're going to talk about Eulerian cycles!

Definition 1

An **Eulerian cycle** or **circuit** / **tour** / **walk** in a digraph G is a closed directed walk on G such that we return to the same point where we start, and every edge is visited exactly once.

We have the following famous result:

Theorem 2 (Euler, 1736)

A digraph G contains an Eulerian cycle if G is connected as an undirected graph and all vertices have equal indegree and outdegree.

We can also say something similar for undirected graphs:

Corollary 3

An undirected graph G contains an (undirected) Eulerian cycle if and only if G is connected and all vertices have even degree.

The story behind these is the bridges of Konigsberg: Euler wanted to use each bridge excatly once. This is pretty easy to prove:

Proof of directed version. It is clear that the conditions are necessary: the graph must be connected, and each edge is entered and exited an equal number of times. To show that this is sufficient, pick some arbitrary edge and keep going until we get stuck at some point (because there are finitely many edges). We can then only get stuck at our initial vertex, because the outdegree is equal to the indegree: for any other vertex, if we enter a times, we should be able to exit a times.

So if we end up at our starting vertex, we are either done, or there are some edges we haven't used. Find a vertex for which an edge hasn't been used: just keep going and repeatedly add edges until everything is used. $\square$

What we're discussing, then, is the next question: how many Eulerian cycles exist? To avoid overcounting, we'll fix our starting edge: basically, we are given a root r and edge e as reference, and we want to see how many Eulerian cycles are distinct up to that. (Since each edge is used exactly once in each Eulerian cycle, it doesn't actually matter which one we use.)

Theorem 4 (B.E.S.T Theorem, 1951)

Let G be a connected digraph, where the indegree of each vertex is equal to its outdegree, and fix a root r. Then the number of Eulerian cycles in G is equal to

$$\#\text{ of Eulerian cycles in } G = \text{In}_r(G) \cdot \prod_{v \text{ vertex}} (\text{outdegree}(v) - 1)!,$$

where $\text{In}_r(G)$ is the number of in-trees rooted at r.

B.E.S.T actually stands for de **B**ruijn, von Aardenne-**E**hrenfest, **S**mith, and **T**utte. Note that we know the value of $\text{In}_r(G)$ from the directed matrix tree theorem!

Fact 5

Since our graph is Eulerian (it satisfies the two requirements), the two Laplacian matrices $L^{\text{in}} = L^{\text{out}}$, and all row and column sums of our Laplacian matrix are zero: this implies that all cofactors of L are the same! Therefore, the number of in-trees rooted at r is the same as the number of out-trees rooted at r, and this is the same for all r.

Example 6

Consider a 4-cycle, where there are a pair of oppositely-directed edges between two opposite vertices in the cycle.

We have two in-trees, and the outdegrees are $2, 1, 2, 1$, so we should have $2 \cdot 1!0!1!0! = 2$ Eulerian cycles.

Proof. Fix our starting root r and starting edge e. Given any Eulerian cycle C, list them in the order they appear in our cycle starting from e: this will order all edges in some manner

$$e_1 e_2 \cdots e_N.$$

This is a total ordering, and it will be denoted by $<_C$.For any vertex $v \neq r$, mark the edge from v which is maximal among all outgoing edges in the order $<_C$: that is, we color the outgoing edge that occurs last in our Eulerian cycle.

Let T be the subgraph formed by those colored edges. We claim that T is an in-tree rooted at r: this follows from the following facts:

- T has $n-1$ edges, where G has n vertices.
- There is no fork: for every vertex, we picked exactly one outgoing edge.
- There are no edges out of the root.
- There are no directed cycles.

This last claim is the least obvious: let's prove it.

Proof. If there were a cycle in T, denoted as $\tilde{C}$, it does not visit the root (because there are no outgoing edges). So let f be the maximal edge in $\tilde{C}$ (by the ordering $<_C$). f is not the last edge in our Eulerian walk (because it doesn't go to the root), so we can let f' be the next edge in our Eulerian cycle C. Meanwhile, define f'' to be the edge that follows f in the cycle $\tilde{C}$.

$f' \neq f''$, because f is supposed to be the largest edge in our cycle, and $f <_C f'$. On the other hand, if f' and f'' are different edges, we know that $f'' <_C f$ by maximality of f, and then $f' > f''$: then we shouldn't have included f'' in T in the first place by maximality of construction of T!. So this is a contradiction, and no such cycle can exist. $\square$

This means that we have a graph with outdegree 1 everywhere except at the root, which is enough to make it an in-tree! So how do we finish the proof? For any vertex v, let w_v be the ordering of all edges outgoing from v except for e and the tree T, given by our order $<_C$. T and e remove 1 from each outdegree, so each w_v corresponds to a permutation of size outdegree$(v) - 1$.

And now the w_vs, plus the intrees, are enough to tell us the whole ordering: we claim this map

$$C \text{ (Eulerian cycle)} \to (T, \{w_v\}_{v \text{ vertex}})$$

is a bijection! To do this, just keep going along, picking the smallest unmarked edge, and only use a marked edge from T if we have no other choice. Then we can never get stuck, because we always use the last edge from each vertex when absolutely necessary by construction! □