Lecture 6: Eulerian Tours

Eric Vigoda
Discrete Mathematics for Computer Science

January 26, 2026

6.1 Inductive Proofs Revisited

A common error in inductive proofs is to "build up" instead of "induct down". Here is an illustration of a wrong proof.

We begin with a false claim that we will "prove" using an incorrect inductive proof.

Conjecture 6.1. *For all $n \geq 2$, for any graph $G = (V, E)$ with $n = |V|$ vertices, the following holds:*

$$\forall v \in V, d(v) \geq 1 \implies G \text{ is connected.} \tag{1}$$

Proof. We will prove Eq. (1) by induction on $n = |V|$.
Base case: The base case is when $n = 2$, in which case the graph $G = (V, E)$ is a single edge since both endpoints have degree ≥ 1. Hence G is connected in this case.
Inductive hypothesis: Assume for integer $k \geq 2$ that Eq. (1) holds for graphs with $n = |V| = k$ vertices.
Inductive step: Take a graph G with k vertices satisfying the inductive hypothesis. Add a new $(k+1)$-st vertex v_{k+1} to G; call the new graph G'. Since v has $d(v) \geq 1$ then connect v to at least one vertex in the original graph G. Since G was connected by the inductive hypothesis, and v is connected to G by at least one edge, then the new graph G' is connected. □

What is the problem in the proof? The conjecture is clearly false so there must be an error in the supposed proof. By building up the graph we are only getting a small subset of possible graphs. We're starting with a single edge. Then in the inductive step we add an edge to it to get a path of length 2. In the next inductive step we add another edge to it to get a tree on 4 vertices. And we keep adding an edge to a tree to get a new tree. So the only graphs we see are trees. Even if we add multiple edges from v_{k+1} then we can get cycles but we don't see all possible graphs.

Consider a graph on $2n$ vertices which has 2 connected components, each of which is a complete graph K_n. Then every vertex has degree $= n-1$ in this graph but it is disconnected. You cannot obtain this graph from the "build up" procedure in the supposed inductive "proof". The construction in the inductive step must be capable of generating any valid graph of size $k+1$ from a valid graph of size k. The correct approach is to consider an arbitrary graph on $k+1$ vertices and then remove a vertex to get a graph on k vertices for which you can apply the inductive hypothesis. Of course, such an approach is sure to fail in this case because the conjecture is false.

6.2 Seven Bridges of Konigsberg

Graphs originated in the work of Leohnard Euler in 1736 to study a classical problem known as the *Seven Bridges of Königsberg*. The city of Königsberg (now called Kaliningrad in Russia) consisted to two islands and two mainland portions of the city, with 7 bridges, see Fig. 1. The problem was to devise a route that starts at one location (either one of the islands or one of the mainlands), crosses each bridge exactly once (in either direction), and ends at the original location.

We can represent the 4 land masses (2 island and 2 mainland portions) by 4 vertices and the 7 bridges by edges between pairs of vertices. In this case there are multiple edges between some pairs of vertices so this is not a graph as we defined it in the first lecture. Technically this is called a multigraph. Let's just

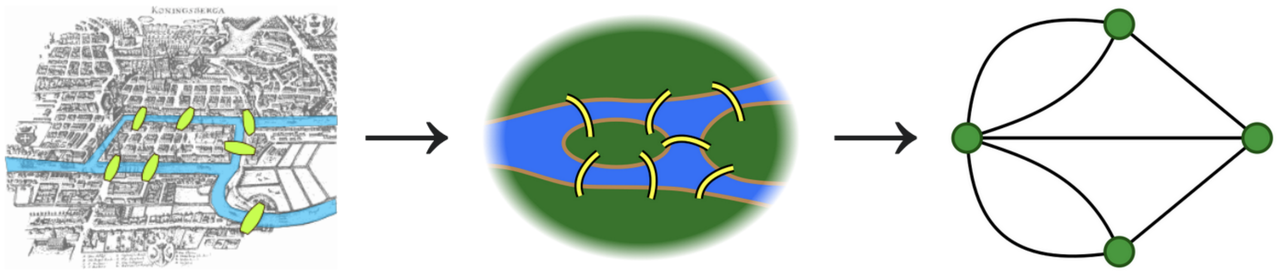

Figure 1: Image of the 7 bridges of Königsberg

consider it for this example and then we'll go back to graphs. Can we start at a vertex v, travel along each edge exactly once and finish at this same vertex v?

For this graph (or technically this multigraph) there is no such tour. Why? Take the top vertex, call it z. The important feature of z is that it has degree 3 – 3 edges are incident to it, by incident we mean that there are 3 edges containing z; two edges with the left vertex and one edge with the right vertex.

Suppose z was our starting/ending location v. Then we take one of the 3 edges out of z to start the tour, at some point we traverse an edge to come back to z, and then we're forced to travel along the third (and last!) edge out of z. But there are no edges that we can use to get back to z at the end. So z can't be the first/last vertex on the tour.

Now suppose instead z was an intermediate vertex on the tour (so not the starting/ending location). Then, at some time we traverse the 1st edge into z and we then travel along the 2nd edge out of z, and finally at some later time we travel along the 3rd (and last!) edge into z. But there are no more edges incident to z so we have no way to travel out of z without revisiting an earlier edge. Hence, z cannot be an intermediate location on the tour.

Therefore, we have shown that there is no tour which visits every edge exactly once for this multigraph. The reason is because there is a vertex of odd degree. This was shown by Euler and hence a tour which visits every edge exactly once is called an **Eulerian tour**. Let's explore when a graph has an Eulerian tour.

6.3 Eulerian tours

In the original 7 bridges problem that Euler solved, we're looking for a walk that visits every edge exactly once and starts and ends at the same vertex. Thus, we say an **Eulerian tour** is a walk $v_0, \ldots, v_\ell$ with the following properties:

- The start and finish vertices are the same: $v_0 = v_\ell$;
- For every edge $\{y, z\} \in E$, there is a unique $0 \leq i < \ell$ where $(v_i = y, v_{i+1} = z)$ or $(v_i = z, v_{i+1} = y)$. By saying there is a unique i we mean that there is exactly one such index i where this holds, and thus every edge is traversed exactly once.

When does a graph have an Eulerian tour? Well certainly a graph has to be connected, otherwise we can't visit every vertex with a single walk. But what if there are isolated vertices; by isolated vertices we mean vertices that have degree 0 so they have no edges incident to them. Then we can't reach isolated vertices by any walk but there is no need to visit them in our tour. So let's keep things simple and restrict attention to connected graphs from now on (for this discussion of Eulerian tours).

From our discussion of the 7 bridges example, we say that a vertex with odd degree appears to rule out the existence of an Eulerian tour. So it appears that all vertices having even degree is a necessary condition for the existence of an Eulerian tour. By necessary condition we mean that for a graph to have an Eulerian tour then every vertex $v \in V$ has $d(v)$ even. So a necessary condition is a precondition for this property. But is it sufficient?

In other words, if all vertices have even degree then does there always exist an Eulerian tour? It turns out that the answer is yes. That means all even degrees is a sufficient condition for an Eulerian tour. Since

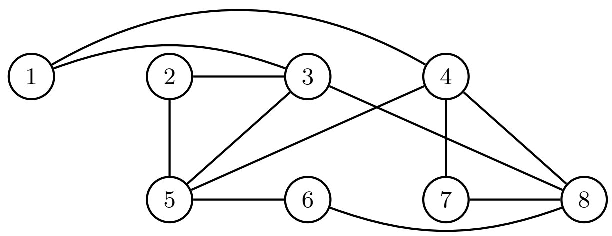

Figure 2: An example graph G, does it have an Eulerian tour?

all even degrees is a necessary and sufficient condition for the existence of an Eulerian tour that means these two properties (namely, all even degrees and Eulerian tour) are equivalent which we can write by an "if and only if" statement. Recall, "iff" or symbolically $\iff$ is used to denote "if and only if".

Lemma 6.2. *For a connected graph $G = (V, E)$,*

$$G \text{ has an Eulerian tour } \iff \forall v \in V,\ d(v) \text{ is even.}$$

Proof. To prove an equivalence (or an iff statement) we need to prove both directions of the implication. Thus, we need to show the following:

$\Rightarrow$: If G has an Eulerian tour then for all vertices $v \in V$, $d(v)$ is even.

$\Leftarrow$: If for all vertices $v \in V$, $d(v)$ is even then G has an Eulerian tour.

Let's start with the forward implication since we already saw the proof idea when discussing the 7 bridges example.

Proof of $\Rightarrow$: Consider an Eulerian tour, call it $T = v_0, \ldots, v_\ell$. Consider a vertex $z \in V$ where z is an intermediate vertex on the tour (not the starting/ending vertex $v_0 = v_\ell$). Consider the times when the tour T visits vertex z, it enters to z along a new edge and leaves from z along a new edge. Thus, there are two distinct edges for every visit to z. Let k denote the number of visits to z on the tour T. Then we've shown that $d(z) = 2k$. Moreover, since G is connected, the tour T visits z at least once so $k \geq 1$, and hence $d(z)$ is even for every intermediate vertex z.

What if z is the starting vertex v_0 and hence also the ending vertex v_ℓ of the tour T? Since T is a tour, which is a cycle with possibly repeated vertices, we can view any other vertex as the starting and ending vertex of the tour and then z is an intermediate vertex and the above argument now applies to z as well.

This proves that if there is an Eulerian tour then every vertex has even degree.

Proof of $\Leftarrow$: Consider a walk $w_0, \ldots, w_k$ where $w_0 = w_k$ then let's call this a **closed walk** (it's a cycle that can repeat vertices); in some texts a closed walk is called a circuit. Let's form a collection of closed walks which are edge disjoint (which means that each closed walk has distinct edges, but a vertex may appear in multiple closed walks or even multiple times in the same closed walk). We will use the following procedure (or algorithm) to form this set of closed walks.

Initially all vertices have even degree. When we use an edge in a closed walk we will remove that edge from future consideration, and we will maintain that all vertices have even degree with respect to the remaining unused edges. Consider an arbitrary vertex z which has some remaining edges. Follow one of the unused edges out of z. Then for each successive vertex y visited, we visit into y along an unused edge and then use another unused edge to exit out of y (there must be another unused as y had even degree for unused edges). Note that we marked two edges incident to y as used so the degree of y for unused edges remains even. Eventually we end up back at the original vertex z (see the next paragraph for an explanation) and this completes the closed walk which starts and ends at z. Note that the initial edge out of z and the final edge back into z give a pair of used edges so the degree of z for unused edges also remains even.

But how do we know we eventually end up back at z. Because we can't get "stuck". If we get stuck at a vertex y where $y \neq z$, then that means we entered a vertex y but had no way to exit y – that can't happen since y has even degree for unused edges, so every time we enter y we have at least one unused edge to use to exit y.

Keep repeating the above procedure on the remaining unused edges, and we end with a collection of closed walks where every edge is in exactly one closed walk, see Fig. 3 for an illustration. If two walks have at least one vertex in common then combine the two closed walks into one bigger closed walk. (More precisely, if closed walk C_1 visits z and closed walk C_2 also visits z, then walk along C_1 until reaching z then traverse C_2, starting/ending at z, and then finish C_1.)

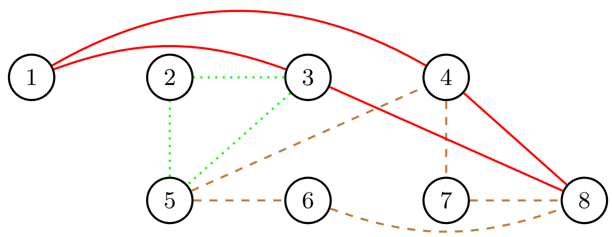

Figure 3: An example graph G with a collection of closed walks (green/dotted, brown/dashed, red/solid).

Notice that all of the closed walks will eventually get combined into one single closed walk – why? Suppose we've combined all the closed walks that we can, and there are at least 2 remaining closed walks, call them $C_1, \ldots, C_k$ for $k \geq 2$. These walks are on disjoint vertices since we cannot combine any pair, and the vertices in C_1 must be disconnected from the vertices in C_2 in the original graph (since there are no edges in any of the walks to connect them), but we assumed the original graph G is connected so this cannot occur. Hence, when we combine the closed walks together we eventually end up with one big closed walk containing all the edges of the original graph G; that gives us our Eulerian tour T.

□

As noted earlier, the above proof of the existence of an Eulerian tour actually provided an algorithm, which can be efficiently implemented, to construct an Eulerian tour. That is how we often prove that a structure exists, by constructing one via an algorithm.

As opposed to an Eulerian tour, one could search for an Eulerian trail if the walk visits every edge exactly once but the start and end vertices are not necessarily the same. In this case, we can have exactly two vertices of odd degree, and these odd degree vertices will be used as the starting and ending vertices of the Eulerian trail.