Today we'll be working with multigraphs, which is one of the very few days we'll be doing so. Recall that a multigraph may have multiple edges between any pair of vertices (some people (usually including myself) also allow multigraphs to have loops, but we will not tolerate this in this class[1]). In a simple graph, an edge is defined by its two end-points, but this is not the case for a multigraph. Additionally, in a multigraph, generally $\deg v \neq |N(v)|$ since $\deg v$ counts the number of edges incident to v. However, facts like the handshaking lemma are still true for multigraphs (the first proof given in class does not work for multigraphs, but the second proof does).

We will be discussing walks in multigraphs. In a simple graph, it was enough to keep track of only the vertices in a walk since edges are dictated by vertices, but, since a pair of vertices may have multiple edges between them in a multigraph, we will need to also keep track of the edges of the walk in a multigraph since we need to know exactly which edges are traversed. To this end, we denote a walk in a multigraph G by

$$(v_0, e_1, v_1, e_2, v_2, \dots, v_{m-1}, e_m, v_m),$$

where $v_0, \dots, v_m$ are vertices, $e_1, \dots, e_m$ are edges and the two end-points of the edge e_i are v_{i-1} and v_i. Note that the length of such a walk is m: the number of edges traversed. In a simple graph, we do not need to explicitly write the e_i's since necessarily $e_i = \{v_{i-1}, v_i\}$.

One other note going forward: We will be working explicitly with multigraphs today, but I will almost always require you only to apply these ideas in the context of simple graphs. So it is okay if you only understand the situation when the graph is simple. There may be rare situations when I require you to work explicitly with multigraphs in these contexts, but, as stated, they will be rare.

Let G be a (multi)graph. A *trail* is a walk in G which traverses each edge *at most* once (but may use a vertex multiple times). A trail is called *closed* if the first and last vertex are equal; otherwise it is called *open*. We also call closed trails *circuits* (in fact, we hardly ever use the term "closed trail", though we will use the term "open trail").

An *Eulerian circuit* is a circuit of G which traverses each edge *exactly* once and uses each vertex at least once.[2] An *Eulerian trail* is an *open* trail of G which traverses each edge *exactly* once and uses each vertex at least once. Note that while a trail could be a circuit, an Eulerian trail can *never* be a circuit by definition (Eulerian trails *must* be open). This is something you just have to get used to. A priori, it could be the case that a (multi)graph has both an Eulerian circuit and an Eulerian trail, but we will soon see that this can never be the case.

Eulerian circuits/trails are named in honor of Leonhard Euler who developed them to solve the "seven bridges of Königsberg problem", which is also considered to be the birth of graph theory as a whole.

[1] Your book refers to multigraphs with loops as "pseudographs". Honestly, I have never heard this term before looking at this book (so it definitely isn't standard), but I guess it's good to have a different term since generally loops throw extra complications into things, e.g. in order to make the handshaking lemma work when loops are allowed, we need to force loops to count twice toward the degree of a vertex. Depending on my mood, we may explicitly work with psuedographs when we get to planarity, but we'll see.

[2] Your book does not require that every vertex appears in an Eulerian circuit, but *I do* since it greatly reduces the complexity of theorem statements and proofs by avoiding annoying case analysis dealing with isolated vertices. Generally, this caveat won't matter in the problems I ask you to solve, but sometimes it will. If I've written a problem where this does indeed matter, please just ask me to clarify and I'll happily do so.

A graph is called *Eulerian* if it contains an Eulerian circuit. We don't have a special name for graphs that have only an Eulerian trail.[3]

Our goal today is to understand exactly when a graph has an Eulerian circuit/trail.

Intuitively, if a graph has an Eulerian circuit, then every vertex must have even degree since every time we "arrive" at a vertex via an edge, we must "leave" that vertex via a different edge. Similarly, if a graph has an Eulerian trail, then the two end-points (which are different here) must have odd degree and the "interior vertices" must have even degree by similar logic.

We formalize this intuition in the following lemma, which is more general since we need this generality later.

Lemma 1. *Let G be a (multi)graph and let*

$$T = (v_0, e_1, v_1, e_2, \dots, v_{m-1}, e_m, v_m)$$

be any trail in G (either open or closed).

1. *If T is closed/a circuit (i.e. $v_0 = v_m$), then T contains an even number of edges incident to v for each $v \in V(G)$.*
2. *If T is open (i.e. $v_0 \neq v_m$), then the following holds:*
   (a) *If $v \in V(G) \setminus \{v_0, v_m\}$, then T contains an even number of edges incident to v.*
   (b) *If $v \in \{v_0, v_m\}$, then T contains an odd number of edges incident to v.*

Proof. Fix any $v \in V(G)$ and let T_v denote the set of edges incident to v contained in the trail T (note that T_v may be the empty set, which has even size). If $m = 0$ (and so $T = (v_0)$, which is a circuit and has no edges), the claim is trivial, so we may suppose that $m \geq 1$.

Define the sets $I_v = \{e_i : i \in [m] \text{ and } v_i = v\}$ (i.e. the edges "going into" v) and $O_v = \{e_i : i \in [m] \text{ and } v_{i-1} = v\}$ (i.e. the edges "coming out of" v). Since T traverses each edge at most once (i.e. $e_i \neq e_j$ for any $i \neq j$), the end-points of e_i and v_{i-1} and v_i and no edge connects a vertex to itself (i.e. $v_{i-1} \neq v_i$), we see that $T_v = I_v \sqcup O_v$. In particular, $|T_v| = |I_v| + |O_v|$.

Now, define the function $f \colon E(T) \to E(T)$ by $f(e_i) = e_{i+1}$ where we interpret $e_{m+1} = e_1$. Certainly f is a bijection since $e_1, \dots, e_m$ are distinct. Additionally $f^{-1}(e_i) = e_{i-1}$ where we interpret $e_0 = e_m$. Note that the function f does note depend on the vertex v, but we will relate f to v next by restricting the domain of f to I_v (and restricting the domain of f^{-1} to O_v).

If $v \notin \{v_0, v_m\}$, then f exhibits a bijection between I_v and O_v since certainly $e_i \in I_v$ if and only if $f(e_i) = e_{i+1} \in O_v$ (we don't need to worry about the "wrap-around" case here). Thus, $|I_v| = |O_v|$ and so $|T_v| = 2|I_v|$, which is even. From here we break into cases:

1. If T is a circuit, then $v_0 = v_m$. So for $v = v_0 = v_m$, we still have that f is a bijection between I_v and O_v since $e_m \in I_v$ and $f(e_m) = e_1 \in O_v$. Thus, again, $|I_v| = |O_v|$ and so $|T_v| = 2|I_v|$, which is even.
2. Otherwise, T is an open trail, so $v_0 \neq v_m$.

[3] Warning: I have seen some people refer to a graph as Eulerian if it has either an Eulerian circuit or an Eulerian trail (though I think this is rare). We definitely won't do so in this class: for us, Eulerian always means that G has an Eulerian circuit. But be aware of this potential issue when you take future classes. (By now, you might be realizing that terminology and notation isn't perfectly standardized across graph theory... Don't worry, this issue only gets worse :P)

(a) If $v = v_m$, then $e_m \in I_v$, yet $f(e_m) = e_1 \notin O_v$. f is thus a bijection between I_v and $O_v \sqcup \{e_1\}$, so $|I_v| = |O_v| + 1 \implies |T_{v_m}| = 2|O_v| + 1$, which is odd.

(b) Similarly, if $v = v_0$, then $e_0 \in O_v$, yet $f^{-1}(e_1) = e_m \notin I_v$. f^{-1} is thus a bijection between O_v and $I_v \sqcup \{e_1\}$, so $|O|_v = |I_v| + 1 \implies |T_{v_0}| = 2|I_v| + 1$, which is odd. □

From here, we find necessary conditions for a graph to have an Eulerian circuit/trail. In order to compress terminology, we say that a graph is *even-regular* if every vertex has even degree.

Corollary 2. *If G is Eulerian (has an Eulerian circuit), then G is connected and even-regular. If G has an Eulerian trail, then G is connected and has* exactly *two odd-degree vertices.*

Proof. Both an Eulerian circuit and trail yield a walk containing all vertices of G, so G must be connected in either case. The degree claims follow immediately from Lemma 1 since an Eulerian circuit/trail sees every edge of G exactly once. □

One thing to note is that the two odd-degree vertices must *always* be the end-points of any Eulerian trail (should it exist).

Corollary 2 is enough to see that the "seven bridges of Königsberg problem" is impossible since the corresponding multigraph has more than two odd-degree vertices.

The amazing thing is that Corollary 2 is actually sufficient! This is one of the many times in graph theory when the "obvious" necessary condition is all that you need (which are always exciting times)! That is to say, G is Eulerian if and only if G is connected and even-regular, and G has an Eulerian trail if and only if G is connected and has exactly two odd-degree vertices!

In order to prove this, we need a lemma which is the crux of the proof.

Lemma 3. *Let G be an even-regular (multi)graph. For any non-isolated vertex $v \in V(G)$, there is a circuit of G which contains v and uses at least one edge (really, at least two edges since it's a circuit).*

Proof. Let $T = (v = v_0, e_1, v_1, e_2, \ldots, v_{m-1}, e_m, v_m)$ be any maximum-length trail (open or closed) starting at v. Since v is not isolated, we know that $m \geq 1$ and so T contains at least one edge. We claim that $v_m = v$ (so T is actually closed and hence a circuit), which will conclude the proof.

Suppose not, then $v_m \neq v = v_0$. Thanks to Lemma 1, we know that T sees an odd number of edges incident to v_m. However, $\deg v_m$ is even, so there must be some edge e_{m+1} incident to v_m which does not appear in T. But then, letting v_{m+1} be the other end-point of e_{m+1}, we find that

$$(v = v_0, e_1, v_1, e_2, \ldots, v_{m-1}, e_m, v_m, e_{m+1}, v_{m+1})$$

is a trail starting at v which is strictly longer than T; a contradiction. □

The moral of the above lemma is that if you start at any non-isolated vertex of an even-regular graph and keep walking (however you wish) until you can no longer continue without repeating an edge, then you'll wind up back where you started.

We are now ready to prove that Corollary 2 is actually biconditional, which, in my opinion, is really cool :) We break this into two steps, first dealing with Eulerian circuits.

The proof is essentially "do the dumb thing and you win". In other words, you can essentially understand the proof as: try your best to build an Eulerian circuit; if ever you get stuck, then you can go back an augment your circuit without undoing any of the work you've already done.

Theorem 4. *Let G be a (multi)graph. Then G is Eulerian (has an Eulerian circuit) if and only if G is connected and G is even-regular.*

Proof. ($\Rightarrow$) Done in Corollary 2.

($\Leftarrow$) To begin, note that the claim is trivial if G has no edges since then $G \cong K_1$ and (v) is an Eulerian circuit (where $V(G) = \{v\}$). So, we may suppose that G has at least one edge. Let

$$C = (v_0, e_1, v_1, e_2, \ldots, v_{m-1}, e_m, v_m = v_0)$$

be a circuit of G of maximum length. If $E(C) = E(G)$, then C is an Eulerian circuit and we are done, so we may suppose that $E(C) \neq E(G)$. Set $G' = G - E(C)$ (but keep all vertices). Lemma 1 implies that C sees an even number of edges incident to each vertex, so we find that G' is also even-regular.

We claim that there is some $v^* \in V(C)$ which is *not* isolated in G'. Indeed, if $V(C) = V(G)$, then this is clear since $E(G) \setminus E(C)$ is non-empty. Otherwise, $V(C) \neq V(G)$ and so both $V(C)$ and $V(G) \setminus V(C)$ are non-empty sets. These sets partition $V(G)$, so since G is connected, there must be some $e \in E(G)$ with one end-point in $V(C)$ and the other in $V(G) \setminus V(C)$. Then $e \in E(G')$ and we can take v^* to be the end-point of e which lives in $V(C)$. Now, v^* is not isolated in G' and G' is even-regular, so Lemma 3 implies that G' contains a circuit

$$(v^* = u_0, s_1, u_2, s_2, \ldots, u_{k-1}, s_k, u_k = v^*)$$

with $k \geq 1$. Note that none of the edges $s_1, \ldots, s_k$ appear in C by construction. Since $v^* \in V(C)$, without loss of generality, we may suppose that $v^* = v_0$ since we can start listing C at any of its vertices. But then

$$(v_0 = v^* = u_0, s_1, u_2, s_2, \ldots, u_{i-1}, s_k, u_k = v^* = v_0, e_1, v_1, \ldots, e_m, v_m = v_0),$$

is a strictly longer circuit than C since $k \geq 1$; a contradiction. □

We finally handle Eulerian trails.

Theorem 5. *Let G be a (multi)graph. Then G has an Eulerian trail if and only if G is connected and has* exactly *two odd-degree vertices.*

Proof. ($\Rightarrow$) Done in Corollary 2.

($\Leftarrow$) Suppose that u and v are the two odd-degree vertices of G. Introduce a new vertex v^* to G which is connected only to u and v; call this new (multi)graph G'. Since G is connected, so is G', and G' is even-regular since u and v were the only two odd-degree vertices of G. By Theorem 4, we know that G' is Eulerian. Fix any Eulerian circuit in G'; we may suppose that it starts and ends at v^*. Then, by deleting v^*, this circuit becomes an Eulerian trail in G with end-points u and v. □

The above proof can also be accomplished by adding an extra edge between u and v. However, if we wished to stay solely within the realm of graphs, this may not be possible since perhaps there is already an edge between u and v. Additionally, this trick of adding a new vertex which is connected to the odd-degree vertices (of which there are an even number) comes in very handy in many other situations (wink wink nudge nudge).