

# Proyecto 4: Hamilton, Euler y Grafos, Parte I

Miembros del Grupo:

Ricardo Castro

Juan Carlos Valverde

Curso: Analisis de Algoritmos

Semestres: II 2025

November 22, 2025

# 1 William Rowan Hamilton

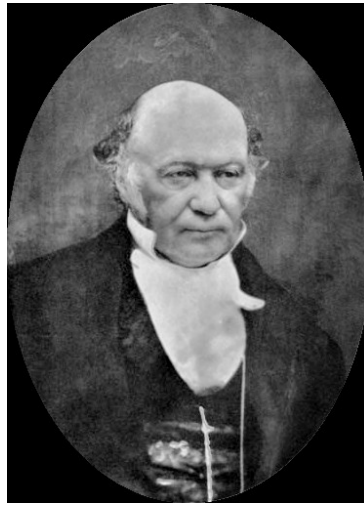


Figure 1: William Rowan Hamilton (1805-1865)

William Rowan Hamilton (1805-1865) fue un matemático, físico y astrónomo irlandés, considerado uno de los científicos más importantes del siglo XIX. Nació en Dublín, Irlanda, y desde muy joven demostró un talento excepcional para las matemáticas y los idiomas. A los 13 años ya dominaba 13 idiomas, incluyendo latín, griego, hebreo, sánscrito y persa.

Hamilton ingresó al Trinity College de Dublín a los 18 años, donde destacó extraordinariamente. A los 22 años, antes de graduarse, fue nombrado Profesor de Astronomía y Director del Observatorio de Dunsink, posiciones que mantendría durante el resto de su vida. Su trabajo abarcó múltiples áreas de las matemáticas y la física.

Una de sus contribuciones más significativas fue el desarrollo del **álgebra de cuaterniones** en 1843. Los cuaterniones son una extensión de los números complejos a cuatro dimensiones, representados como  $q = a + bi + cj + dk$ , donde  $i$ ,  $j$  y  $k$  son unidades imaginarias que satisfacen relaciones específicas. Esta invención tuvo un impacto profundo en la física, especialmente en la mecánica cuántica y la computación gráfica moderna.

En el campo de la mecánica, Hamilton desarrolló el **principio de Hamilton**, también conocido como principio de mínima acción, que reformuló la mecánica clásica en términos de variaciones. Este principio es fundamental en la física teórica y proporciona una base elegante para la mecánica lagrangiana y hamiltoniana.

En teoría de grafos, Hamilton es conocido por el **problema del ciclo hamiltoniano**, que formuló en 1857. El problema consiste en determinar si existe un ciclo en un grafo que visite cada vértice exactamente una vez y regrese al vértice inicial. Este problema, aunque aparentemente simple, resultó ser NP-completo y ha sido objeto de extensa investigación en ciencias de la computación.

Hamilton también hizo importantes contribuciones a la óptica geométrica, desarrollando una teoría matemática de los rayos de luz que fue precursora de la mecánica cuántica. Su trabajo en óptica estableció conexiones profundas entre la física y las matemáticas.

A lo largo de su carrera, Hamilton publicó numerosos artículos y mantuvo correspondencia con los principales científicos de su época, incluyendo a John Herschel y Peter Guthrie Tait. Su legado perdura no solo en las matemáticas y la física, sino también en la forma en que concebimos las estructuras algebraicas y los problemas de optimización.

## 2 Ciclos y Rutas Hamiltonianas

### 2.1 Definiciones Fundamentales

Un **ciclo hamiltoniano** es un ciclo cerrado en un grafo que visita cada vértice exactamente una vez y regresa al vértice inicial. Formalmente, dado un grafo  $G = (V, E)$  con  $n$  vértices, un ciclo hamiltoniano es una secuencia de vértices  $v_1, v_2, \dots, v_n, v_1$  tal que:

- Cada vértice aparece exactamente una vez en la secuencia (excepto el inicial que aparece al inicio y al final)
- Para cada par consecutivo  $(v_i, v_{i+1})$  en la secuencia, existe una arista  $(v_i, v_{i+1}) \in E$
- Existe una arista  $(v_n, v_1) \in E$  que cierra el ciclo

Una **ruta hamiltoniana** (o camino hamiltoniano) es un camino simple que visita cada vértice exactamente una vez, pero no necesariamente regresa al punto de partida. Es decir, es una secuencia de vértices  $v_1, v_2, \dots, v_n$  donde cada vértice aparece exactamente una vez y cada par consecutivo está conectado por una arista.

### 2.2 Importancia y Aplicaciones

El problema de determinar si un grafo tiene un ciclo o ruta hamiltoniana es uno de los 21 problemas NP-completos originales identificados por Karp en 1972. Esto significa que:

- No se conoce un algoritmo eficiente (polinomial) que resuelva el problema para grafos arbitrarios
- Si existiera tal algoritmo, se resolverían todos los problemas NP-completos
- Los algoritmos actuales tienen complejidad exponencial en el peor caso

A pesar de su complejidad, los ciclos y rutas hamiltonianas tienen numerosas aplicaciones prácticas:

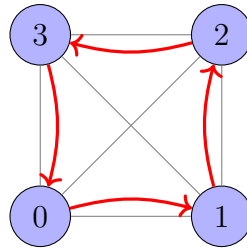
- **Problema del viajante (TSP):** Encontrar la ruta más corta que visite todas las ciudades exactamente una vez
- **Secuenciación de tareas:** Optimizar el orden de ejecución de tareas con dependencias
- **Diseño de circuitos:** Encontrar rutas que pasen por todos los puntos de un circuito
- **Análisis de redes:** Estudiar la conectividad y estructura de redes complejas

### 2.3 Ejemplos Visuales

A continuación se presentan ejemplos gráficos que ilustran estos conceptos:

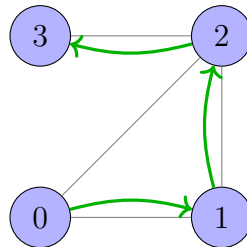
### 2.3.1 Ejemplo 1: Ciclo Hamiltoniano

El siguiente grafo muestra un ejemplo de ciclo hamiltoniano. El ciclo está marcado en rojo y sigue la secuencia:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$ .



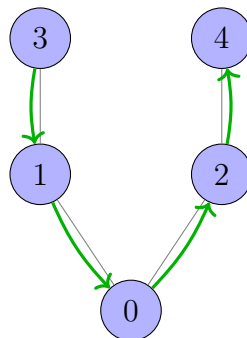
### 2.3.2 Ejemplo 2: Ruta Hamiltoniana

El siguiente grafo muestra una ruta hamiltoniana (no un ciclo, ya que no regresa al vértice inicial). La ruta está marcada en verde y sigue:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ .



### 2.3.3 Ejemplo 3: Grafo sin Ciclo Hamiltoniano

No todos los grafos tienen ciclos hamiltonianos. El siguiente grafo (un árbol) no tiene ciclo hamiltoniano porque es acíclico, pero sí tiene una ruta hamiltoniana.



## 2.4 Algoritmos y Complejidad

En este proyecto, se utiliza un algoritmo de **backtracking** (vuelta atrás) para determinar si existe al menos un ciclo o ruta hamiltoniana en el grafo. El algoritmo explora sistemáticamente todas las posibles rutas, retrocediendo cuando una ruta parcial no puede completarse.

La complejidad temporal del algoritmo de backtracking para este problema es  $O(n!)$  en el peor caso, donde  $n$  es el número de vértices. Esto se debe a que, en el peor escenario, debe explorar todas las permutaciones posibles de los vértices.

Aunque el algoritmo implementado determina la *existencia* de un ciclo o ruta hamiltoniana, no encuentra la solución específica. Para encontrar la solución completa, sería necesario

modificar el algoritmo para almacenar y retornar la secuencia de vértices que forma el ciclo o ruta.

### 3 Leonhard Euler



Figure 2: Leonhard Euler (1707-1783)

Leonhard Euler (1707-1783) fue un matemático y físico suizo considerado uno de los matemáticos más prolíficos e influyentes de la historia. Nació en Basilea, Suiza, y desde muy joven mostró un talento excepcional para las matemáticas. Su padre, Paul Euler, era pastor calvinista y quería que su hijo siguiera sus pasos, pero el joven Leonhard estaba destinado a convertirse en una de las mentes más brillantes de la ciencia.

Euler estudió en la Universidad de Basilea bajo la tutela de Johann Bernoulli, uno de los matemáticos más destacados de su época. A los 19 años, Euler ya había completado su maestría y comenzaba a publicar trabajos matemáticos. En 1727, a los 20 años, fue invitado a unirse a la Academia de Ciencias de San Petersburgo, donde permanecería hasta 1741, y luego regresaría en 1766 hasta su muerte.

La productividad de Euler fue extraordinaria. Publicó más de 800 artículos y libros durante su vida, y su obra completa, que aún se está compilando, se estima que comprenderá más de 80 volúmenes. Sus contribuciones abarcan prácticamente todas las áreas de las matemáticas: análisis matemático, teoría de números, geometría, álgebra, mecánica, óptica, astronomía y música.

Una de sus contribuciones más fundamentales fue el desarrollo y sistematización del **cálculo infinitesimal**. Euler introdujo la notación matemática moderna que usamos hoy en día, incluyendo el símbolo  $e$  para la base del logaritmo natural (aproximadamente 2.71828), el símbolo  $i$  para la unidad imaginaria, y la notación  $f(x)$  para funciones. También estableció la famosa identidad  $e^{i\pi} + 1 = 0$ , conocida como la identidad de Euler, que conecta cinco de los números más importantes en matemáticas.

En 1736, Euler resolvió el **problema de los puentes de Königsberg**, que se considera el origen de la teoría de grafos. El problema consistía en determinar si era posible cruzar los siete puentes de la ciudad de Königsberg (hoy Kaliningrado) exactamente una vez y regresar al punto de partida. Euler demostró que esto era imposible, sentando las bases para lo que hoy conocemos como grafos eulerianos.

Euler hizo contribuciones fundamentales a la **teoría de números**, incluyendo el teorema de Euler en aritmética modular, que generaliza el pequeño teorema de Fermat.

También trabajó extensamente en la función zeta de Riemann (aunque no con ese nombre), estableciendo la relación entre los números primos y los números naturales.

En **mecánica**, Euler desarrolló las ecuaciones de movimiento de Euler para cuerpos rígidos y contribuyó significativamente a la mecánica de fluidos. Sus trabajos en óptica y astronomía también fueron pioneros, incluyendo cálculos precisos de órbitas planetarias y el desarrollo de la teoría de la refracción de la luz.

A pesar de perder la visión de un ojo en 1735 y quedarse completamente ciego en 1766, Euler continuó trabajando con una productividad asombrosa. Su memoria prodigiosa le permitía realizar cálculos complejos mentalmente, y dictaba sus trabajos a sus hijos y asistentes. De hecho, algunos de sus trabajos más importantes fueron producidos durante su ceguera total.

El legado de Euler es inmenso. Muchos conceptos, teoremas y fórmulas llevan su nombre: el número de Euler ( $e$ ), la fórmula de Euler, el método de Euler para ecuaciones diferenciales, los ángulos de Euler, la función phi de Euler, y muchos más. Su influencia se extiende no solo a las matemáticas puras, sino también a la física, la ingeniería y la computación moderna.

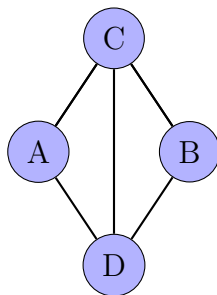
Euler murió en San Petersburgo en 1783, dejando un legado que continúa inspirando a matemáticos y científicos hasta el día de hoy. Su enfoque sistemático, su notación clara y su capacidad para encontrar conexiones profundas entre diferentes áreas del conocimiento lo convierten en uno de los pilares fundamentales de las matemáticas modernas.

## 4 Ciclos y Rutas Eulerianas

### 4.1 El Problema de los Puentes de Königsberg

El origen de los ciclos eulerianos se remonta al famoso **problema de los puentes de Königsberg**, resuelto por Euler en 1736. La ciudad de Königsberg (hoy Kaliningrado) estaba dividida por el río Pregel en cuatro regiones conectadas por siete puentes. El problema consistía en determinar si era posible dar un paseo que cruzara cada puente exactamente una vez y regresar al punto de partida.

Euler modeló este problema como un grafo, donde cada región era un vértice y cada puente era una arista. Demostró que tal recorrido era imposible, estableciendo así los fundamentos de la teoría de grafos y los grafos eulerianos.



Grafo de los puentes de Königsberg  
Grados:  $A=3$ ,  $B=3$ ,  $C=5$ ,  $D=3$  (todos impares)

Como todos los vértices tienen grado impar, el grafo no puede tener un ciclo euleriano. Este fue el primer resultado en teoría de grafos.

## 4.2 Definiciones Fundamentales

Un **ciclo euleriano** es un ciclo cerrado en un grafo que recorre cada arista exactamente una vez y regresa al vértice inicial. Formalmente, dado un grafo  $G = (V, E)$  con  $m$  aristas, un ciclo euleriano es una secuencia de vértices  $v_1, v_2, \dots, v_k, v_1$  tal que:

- Cada arista del grafo aparece exactamente una vez en el ciclo
- El ciclo comienza y termina en el mismo vértice
- Cada par consecutivo de vértices está conectado por una arista

Un **camino euleriano** (o ruta euleriana) es un camino que recorre cada arista exactamente una vez, pero no necesariamente regresa al punto de partida. Es decir, es una secuencia de vértices  $v_1, v_2, \dots, v_k$  donde cada arista aparece exactamente una vez.

Un grafo es **euleriano** si tiene un ciclo euleriano. Un grafo es **semieuleriano** si tiene un camino euleriano pero no un ciclo euleriano.

## 4.3 Teorema de Euler

Euler estableció condiciones necesarias y suficientes para la existencia de ciclos y caminos eulerianos:

**Teorema (Euler, 1736):** Para un grafo no dirigido conexo  $G$ :

- $G$  tiene un ciclo euleriano si y solo si todos los vértices tienen grado par
- $G$  tiene un camino euleriano (pero no un ciclo) si y solo si tiene exactamente dos vértices de grado impar. Estos vértices serán el inicio y el final del camino

Para grafos dirigidos:

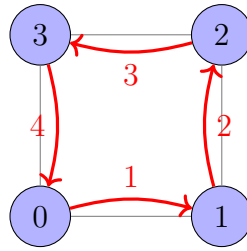
- Un grafo dirigido tiene un ciclo euleriano si y solo si es fuertemente conexo y cada vértice tiene el mismo grado de entrada que de salida
- Un grafo dirigido tiene un camino euleriano si tiene exactamente un vértice con  $\deg_{\text{salida}} - \deg_{\text{entrada}} = 1$  (inicio), exactamente un vértice con  $\deg_{\text{entrada}} - \deg_{\text{salida}} = 1$  (fin), y todos los demás tienen grados iguales

## 4.4 Ejemplos Visuales

A continuación se presentan ejemplos gráficos que ilustran estos conceptos:

### 4.4.1 Ejemplo 1: Grafo Euleriano (Ciclo Euleriano)

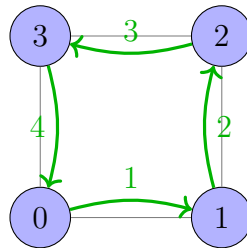
El siguiente grafo es euleriano porque todos los vértices tienen grado par (2 o 4). El ciclo euleriano está marcado en rojo y recorre todas las aristas exactamente una vez.



Ciclo:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$   
 Grados: todos pares (2)

#### 4.4.2 Ejemplo 2: Grafo Semieuleriano (Camino Euleriano)

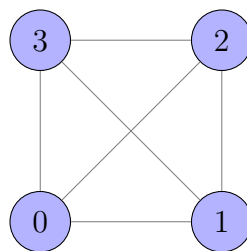
El siguiente grafo es semieuleriano porque tiene exactamente dos vértices de grado impar (vértices 0 y 3, ambos con grado 3). El camino euleriano está marcado en verde y comienza en el vértice 0 y termina en el vértice 3.



Camino:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$   
 Grados:  $0=3$ ,  $1=2$ ,  $2=2$ ,  $3=3$  (dos impares)

#### 4.4.3 Ejemplo 3: Grafo No Euleriano

El siguiente grafo no es euleriano ni semieuleriano porque tiene más de dos vértices de grado impar (vértices 0, 1, 2 y 3, todos con grado 3). Por lo tanto, no existe ningún camino que recorra todas las aristas exactamente una vez.

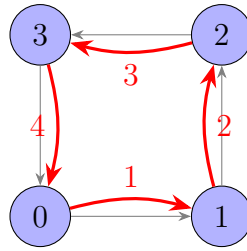


Grados: todos impares (3)  
 No es euleriano ni semieuleriano

#### 4.4.4 Ejemplo 4: Grafo Dirigido Euleriano

En grafos dirigidos, un ciclo euleriano requiere que cada vértice tenga el mismo grado de entrada que de salida. El siguiente ejemplo muestra un grafo dirigido euleriano.





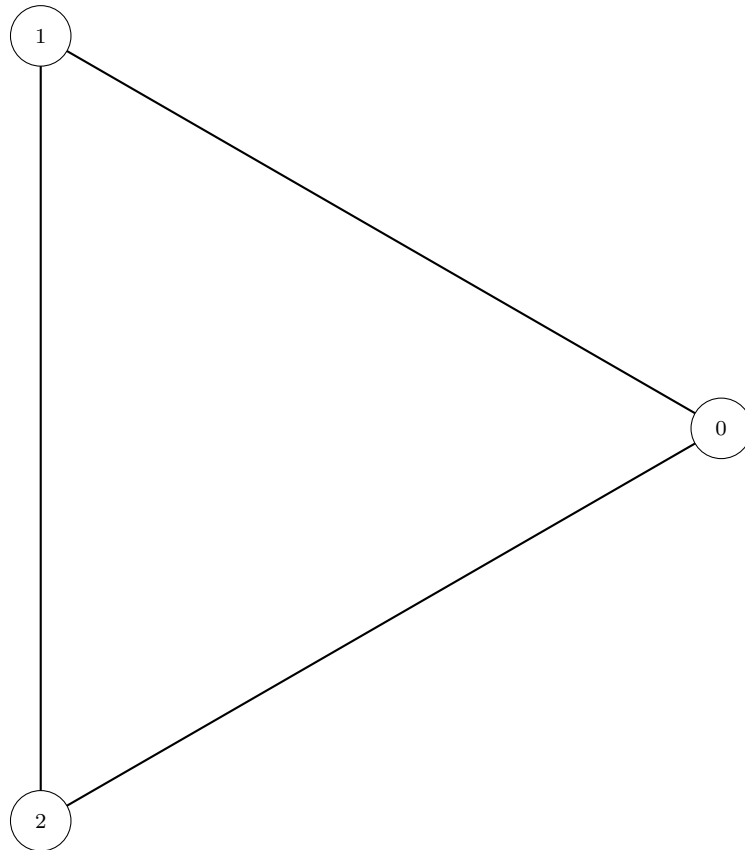
Ciclo:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$   
 Grados: entrada = salida para todos (1)

## 4.5 Aplicaciones Prácticas



Los ciclos y caminos eulerianos tienen numerosas aplicaciones en la vida real:

- **Recolección de basura:** Optimizar rutas para que los camiones pasen por todas las calles exactamente una vez
- **Inspección de redes:** Verificar todas las conexiones de una red (eléctrica, de agua, de datos) de manera eficiente
- **Impresión de circuitos:** Encontrar rutas que pasen por todas las líneas de un circuito impreso sin repetición
- **Postal y entrega:** Diseñar rutas de entrega que cubran todas las calles sin duplicar esfuerzo
- **Análisis de ADN:** Reconstruir secuencias genéticas a partir de fragmentos

## 5 Grafo Original



### 5.1 Leyenda de Colores

-  Nodos de grado impar
-  Nodos de grado par

## 6 Propiedades del Grafo

### 6.1 Ciclos y Rutas Hamiltonianas

Para determinar la existencia de ciclos y rutas hamiltonianas en este grafo, se ha utilizado un algoritmo de backtracking que explora sistemáticamente todas las posibles secuencias de vértices. El algoritmo verifica si existe al menos una permutación de los vértices que forme un ciclo o ruta válida según las aristas presentes en el grafo.

**Resultado: El grafo contiene al menos un ciclo hamiltoniano.**

Esto significa que existe al menos una secuencia de vértices  $v_1, v_2, \dots, v_n, v_1$  tal que:

- Cada vértice del grafo aparece exactamente una vez en la secuencia (excepto el vértice inicial que aparece al inicio y al final)
- Cada par consecutivo de vértices en la secuencia está conectado por una arista
- El último vértice está conectado al primero, cerrando el ciclo

La existencia de un ciclo hamiltoniano indica que es posible visitar todos los vértices del grafo exactamente una vez y regresar al punto de partida, siguiendo únicamente las aristas existentes. Esta propiedad es de gran importancia en problemas de optimización como el problema del viajante (TSP) y en el diseño de circuitos.

**Resultado: El grafo contiene al menos una ruta hamiltoniana.**

Esto significa que existe al menos una secuencia de vértices  $v_1, v_2, \dots, v_n$  tal que:

- Cada vértice del grafo aparece exactamente una vez en la secuencia
- Cada par consecutivo de vértices en la secuencia está conectado por una arista
- El camino no necesariamente regresa al vértice inicial

La existencia de una ruta hamiltoniana indica que es posible visitar todos los vértices del grafo exactamente una vez, aunque no se regrese al punto de partida. Esta propiedad es útil en problemas de secuenciación, donde se necesita un orden específico de elementos sin repetición.

## 6.2 Propiedades Eulerianas

Para determinar las propiedades eulerianas del grafo, se ha analizado la paridad de los grados de los vértices. Según el teorema de Euler (1736), las condiciones para la existencia de ciclos y caminos eulerianos dependen directamente de los grados de los vértices, lo que hace que este problema sea computacionalmente más simple que el problema hamiltoniano.

**Resultado: El grafo es euleriano.**

Esto significa que el grafo contiene un **ciclo euleriano**, es decir, un ciclo cerrado que recorre cada arista del grafo exactamente una vez y regresa al vértice inicial. Esta es una propiedad muy deseable en aplicaciones prácticas como la optimización de rutas de recolección, inspección de redes y diseño de circuitos.

**Análisis para grafo no dirigido:**

Para que un grafo no dirigido sea euleriano, según el teorema de Euler, deben cumplirse dos condiciones:

1. El grafo debe ser **conexo**: todos los vértices deben estar conectados entre sí, de manera que exista un camino entre cualquier par de vértices.
2. Todos los vértices deben tener **grado par**: cada vértice debe estar conectado a un número par de aristas.

En este grafo, se ha verificado que ambas condiciones se cumplen. La condición de grados pares es necesaria porque, en un ciclo euleriano, cada vez que se entra a un vértice por una arista, se debe salir por otra arista diferente. Por lo tanto, cada vértice debe tener un número par de aristas incidentes.

**Implicaciones prácticas:**

La existencia de un ciclo euleriano significa que es posible diseñar una ruta óptima que recorra todas las aristas del grafo exactamente una vez, sin necesidad de repetir ninguna conexión. Esto es especialmente valioso en aplicaciones donde se busca minimizar el tiempo o costo de recorrer todas las conexiones de una red.

## 7 Ciclo o Ruta Hamiltoniana

### Ciclo Hamiltoniano encontrado:

El grafo contiene un ciclo hamiltoniano. A continuación se presenta una secuencia de vértices que forma dicho ciclo:

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 0$$

Esta secuencia visita cada vértice exactamente una vez (excepto el vértice inicial que aparece al inicio y al final) y forma un ciclo cerrado.

## 8 Carl Hierholzer



Figure 3: Carl Hierholzer (1840-1871)

Carl Hierholzer (1840-1871) fue un matemático alemán conocido principalmente por su contribución a la teoría de grafos, específicamente por el desarrollo del algoritmo que lleva su nombre para encontrar ciclos eulerianos en grafos.

Hierholzer nació en Karlsruhe, Alemania, y estudió matemáticas en la Universidad de Heidelberg. Aunque su carrera fue relativamente corta debido a su temprana muerte a los 31 años, dejó una contribución significativa a las matemáticas.

En 1873, dos años después de su muerte, se publicó su trabajo más importante: *Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren* (Sobre la posibilidad de recorrer un trazo de líneas sin repetición y sin interrupción). En este trabajo, Hierholzer presentó un algoritmo eficiente para encontrar ciclos eulerianos en grafos, resolviendo de manera práctica el problema que Euler había caracterizado teóricamente más de un siglo antes.

El **algoritmo de Hierholzer** es notable por su elegancia y eficiencia. A diferencia de otros métodos, este algoritmo tiene complejidad temporal  $O(m)$ , donde  $m$  es el número de aristas, lo que lo hace óptimo para este problema. El algoritmo funciona construyendo ciclos parciales y luego empalmándolos para formar el ciclo euleriano completo.

Aunque Hierholzer murió antes de ver su trabajo publicado, su algoritmo se convirtió en uno de los métodos estándar para resolver problemas eulerianos y sigue siendo ampliamente utilizado en la actualidad en aplicaciones de optimización de rutas, diseño de circuitos y análisis de redes.

## 9 Ciclo Euleriano con Hierholzer

Se ha ejecutado el algoritmo de Hierholzer para encontrar un ciclo euleriano en el grafo. El algoritmo funciona de la siguiente manera:

1. Comienza en un vértice arbitrario con aristas incidentes
2. Construye un ciclo aleatorio hasta que no se puedan agregar más aristas
3. Si quedan aristas sin visitar, encuentra un vértice en el ciclo actual que tenga aristas sin visitar y construye un nuevo ciclo desde ese vértice
4. Empalma el nuevo ciclo con el ciclo principal
5. Repite hasta que todas las aristas hayan sido visitadas

### Ciclo Euleriano encontrado:

A continuación se presenta la secuencia de vértices que forma el ciclo euleriano (los vértices pueden aparecer múltiples veces, ya que se recorren todas las aristas):

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 0$$

Este ciclo recorre cada arista del grafo exactamente una vez y regresa al vértice inicial.

### 9.1 Trabajo Extra Opcional: Ejecución Paso a Paso del Algoritmo de Hierholzer

A continuación se presenta una visualización detallada de cómo el algoritmo de Hierholzer encuentra el ciclo euleriano. El algoritmo construye ciclos parciales y los empalma para formar la solución completa.

#### Descripción del Algoritmo:

El algoritmo de Hierholzer funciona de la siguiente manera:

1. Se inicia desde un vértice arbitrario con aristas incidentes
2. Se construye un ciclo aleatorio recorriendo aristas no visitadas hasta que no se puedan agregar más aristas (se regresa al vértice inicial del ciclo)
3. Si quedan aristas sin visitar, se encuentra un vértice en el ciclo actual que tenga aristas sin visitar y se construye un nuevo ciclo desde ese vértice
4. Se empalma el nuevo ciclo con el ciclo principal insertándolo en el punto donde se encontró el vértice con aristas pendientes
5. Se repite el proceso hasta que todas las aristas hayan sido visitadas

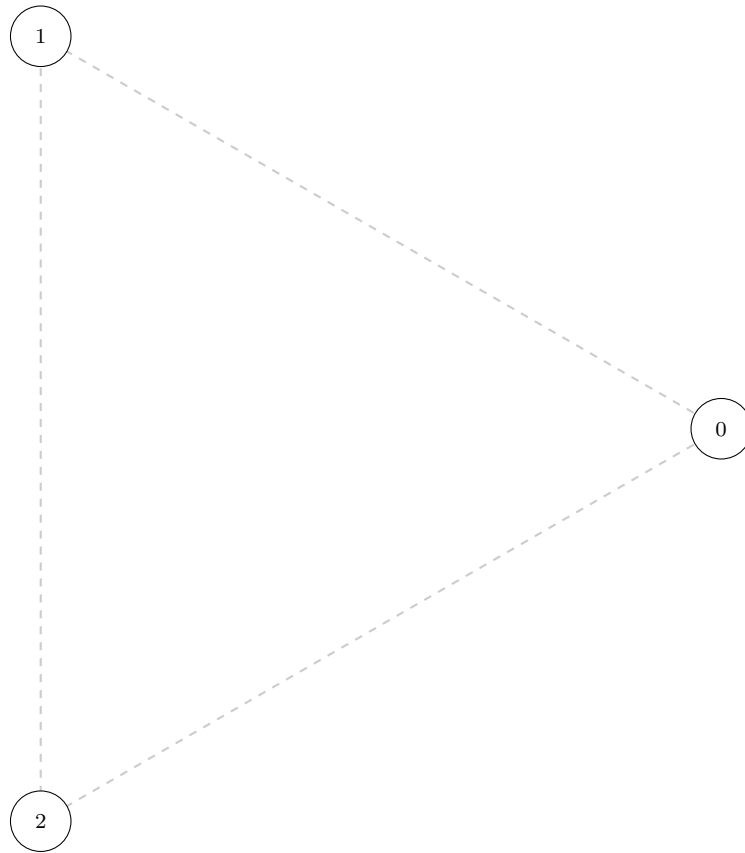
#### Ejecución Paso a Paso:

A continuación se muestra la ejecución detallada del algoritmo. En cada paso se muestra el estado del grafo, donde:

- Las aristas en gris punteado son aristas que aún no se han recorrido
- Las aristas en rojo grueso forman el ciclo parcial que se está construyendo actualmente
- Las aristas en otros colores (azul, verde, naranja, etc.) son ciclos parciales que ya se completaron

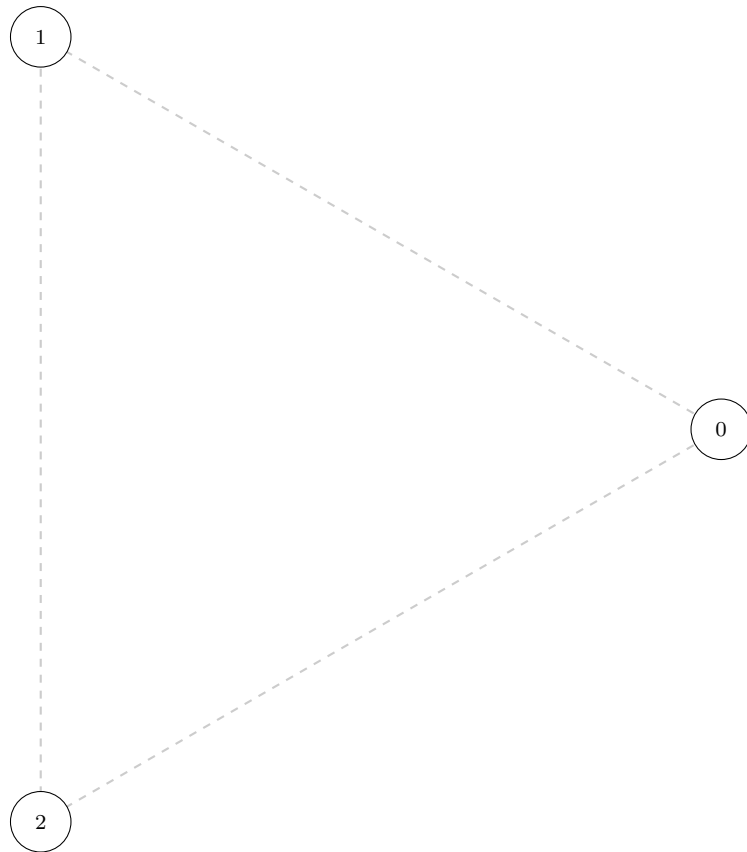
### 9.1.1 Paso 1

Inicio del algoritmo. Se comienza desde el vértice 0.



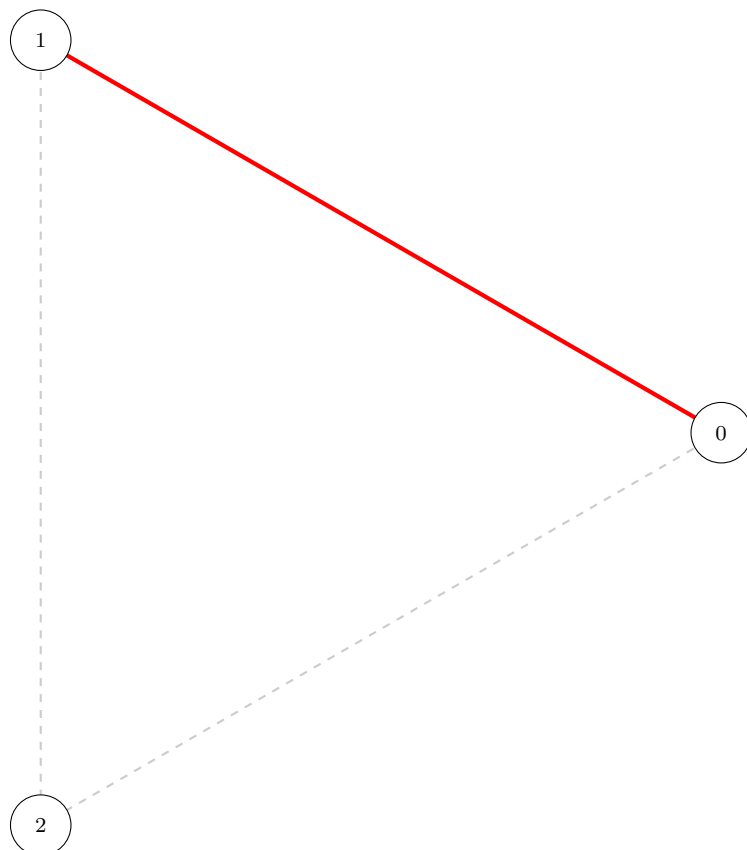
### 9.1.2 Paso 2

Se agrega la arista  $0 \rightarrow 1$ . Se continúa construyendo el ciclo parcial.



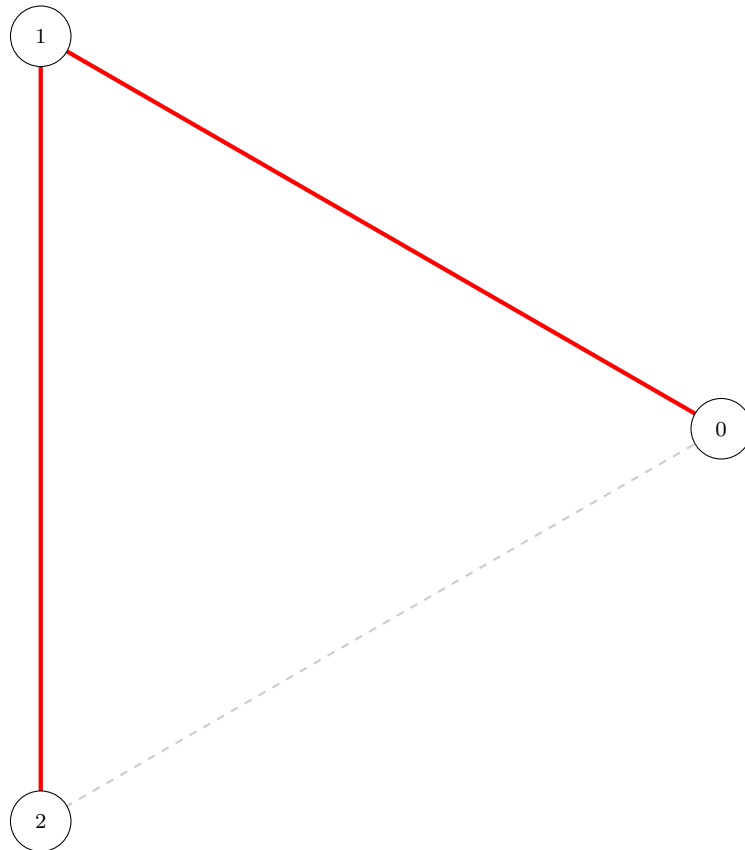
### 9.1.3 Paso 3

Se agrega la arista  $1 \rightarrow 2$ . Se continúa construyendo el ciclo parcial.



#### 9.1.4 Paso 4

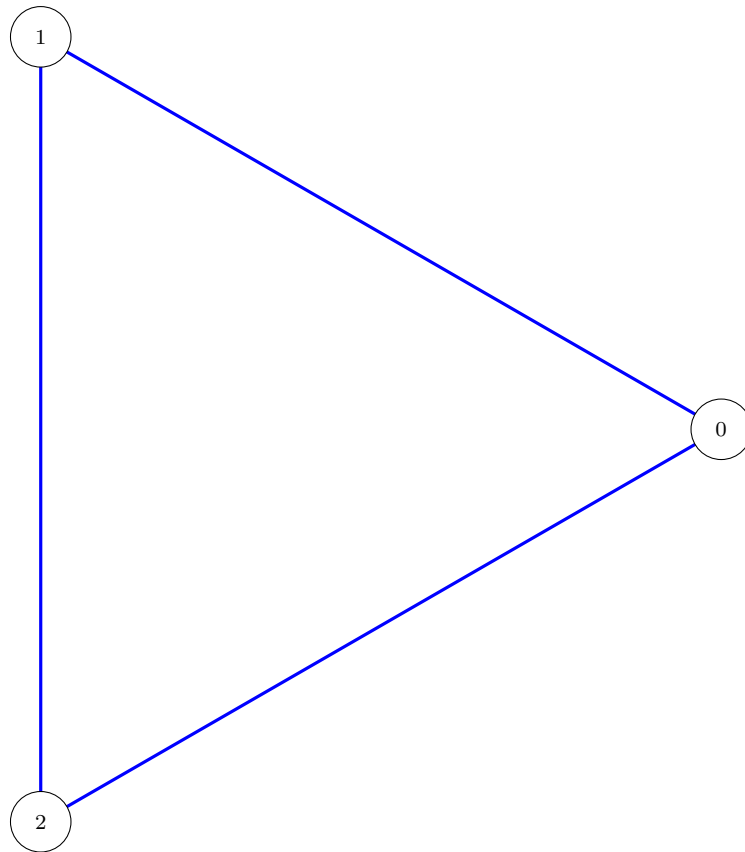
Se agrega la arista  $2 \rightarrow 0$ . Se continúa construyendo el ciclo parcial.



#### 9.1.5 Paso 5

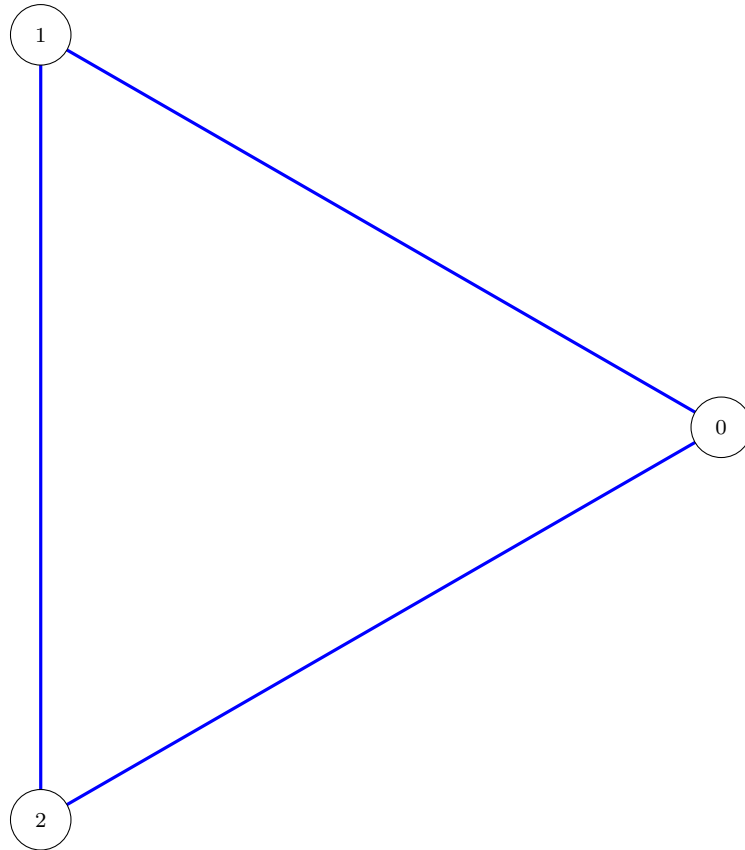
Se completa un ciclo parcial que termina en el vértice 0. Este es el primer ciclo encontrado.





#### 9.1.6 Paso 6

Se completa un ciclo parcial que termina en el vértice 0. Este ciclo se empalmará con los ciclos anteriores.

**Ciclo Euleriano Final:**

El algoritmo encontró el siguiente ciclo euleriano completo:

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 0$$

**Análisis de la Ejecución:**

El ciclo encontrado tiene 4 vértices (algunos pueden repetirse ya que se recorren todas las aristas). El algoritmo garantiza que cada arista se recorra exactamente una vez y que se regrese al vértice inicial, formando así un ciclo euleriano completo.

**Complejidad:** El algoritmo de Hierholzer tiene complejidad temporal  $O(m)$ , donde  $m$  es el número de aristas, lo que lo hace óptimo para este problema.

## 10 Pierre-Henry Fleury

Pierre-Henry Fleury fue un matemático francés del siglo XIX conocido por su contribución al desarrollo del algoritmo que lleva su nombre para encontrar ciclos y caminos eulerianos en grafos.

Aunque se conoce menos sobre la vida personal de Fleury en comparación con otros matemáticos de la época, su trabajo en teoría de grafos ha tenido un impacto significativo. El algoritmo de Fleury fue desarrollado como una alternativa al algoritmo de Hierholzer, ofreciendo un enfoque diferente para resolver el mismo problema.

El **algoritmo de Fleury** se caracteriza por su enfoque de eliminación de aristas. A diferencia de Hierholzer, que construye ciclos y los empalma, Fleury trabaja eliminando aristas del grafo mientras construye el ciclo o camino euleriano. La clave del algoritmo está en la regla de selección de aristas: siempre que sea posible, se debe evitar elegir una



Figure 4: Pierre-Henry Fleury (siglo XIX)

arista que sea un *punte* (una arista cuya eliminación desconecte el grafo), a menos que no haya otra opción.

El algoritmo de Fleury tiene complejidad temporal  $O(m^2)$  en el peor caso, donde  $m$  es el número de aristas, debido a la necesidad de verificar si una arista es un puente en cada paso. Aunque es menos eficiente que el algoritmo de Hierholzer, ofrece una perspectiva diferente y es útil para entender la estructura de los grafos eulerianos.

El trabajo de Fleury, junto con el de Hierholzer, proporcionó herramientas prácticas para resolver problemas que Euler había caracterizado teóricamente, permitiendo la aplicación de estos conceptos en problemas reales de optimización de rutas y diseño de redes.

## 11 Trabajo Extra Opcional: Ejecución Paso a Paso del Algoritmo de Fleury para Ciclos Eulerianos

Este grafo es euleriano, por lo que contiene un ciclo euleriano. A continuación se presenta una visualización detallada de cómo el algoritmo de Fleury encuentra el ciclo euleriano eliminando aristas del grafo.

### **Descripción del Algoritmo:**

El algoritmo de Fleury funciona de la siguiente manera:

1. Se comienza en un vértice arbitrario del grafo
2. En cada paso, se selecciona una arista incidente al vértice actual
3. Se verifica si la arista es un *punte* (una arista cuya eliminación desconectaría el grafo). Si es posible, se evita seleccionar puentes, a menos que sea la única opción disponible
4. Se elimina la arista seleccionada del grafo
5. Se mueve al vértice conectado por esa arista
6. Se repite el proceso hasta que todas las aristas hayan sido eliminadas

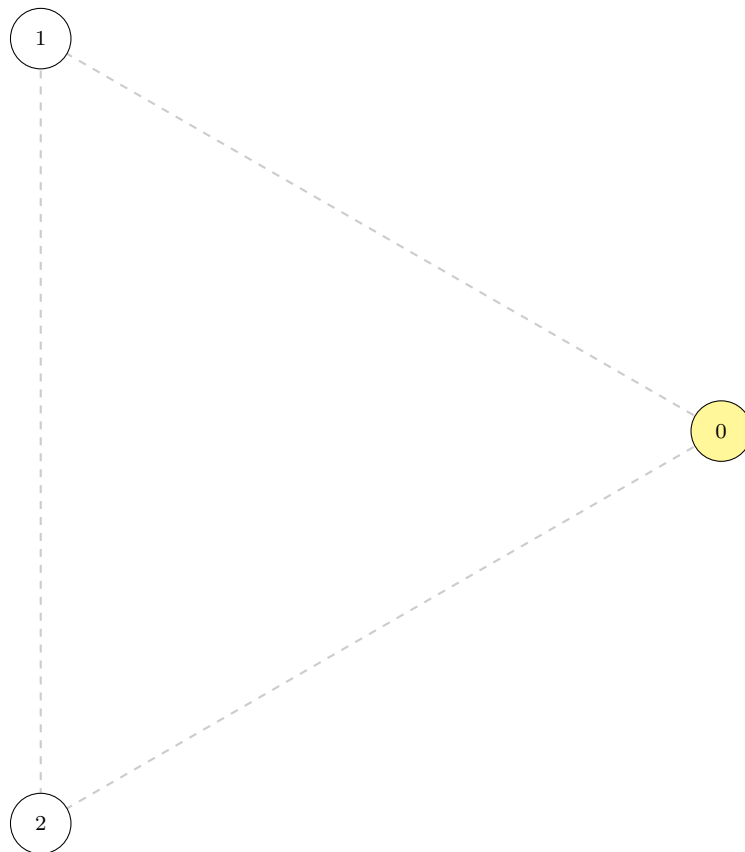
### Ejecución Paso a Paso:

A continuación se muestra la ejecución detallada del algoritmo. En cada paso se muestra el estado del grafo, donde:

- Las aristas en gris punteado son aristas que aún no se han eliminado
- Las aristas en azul grueso forman la ruta construida hasta el momento
- La arista en verde grueso es la arista elegida en este paso (no es puente)
- La arista en rojo grueso es la arista elegida en este paso (es un puente, pero es la única opción)
- El vértice en amarillo es el vértice actual

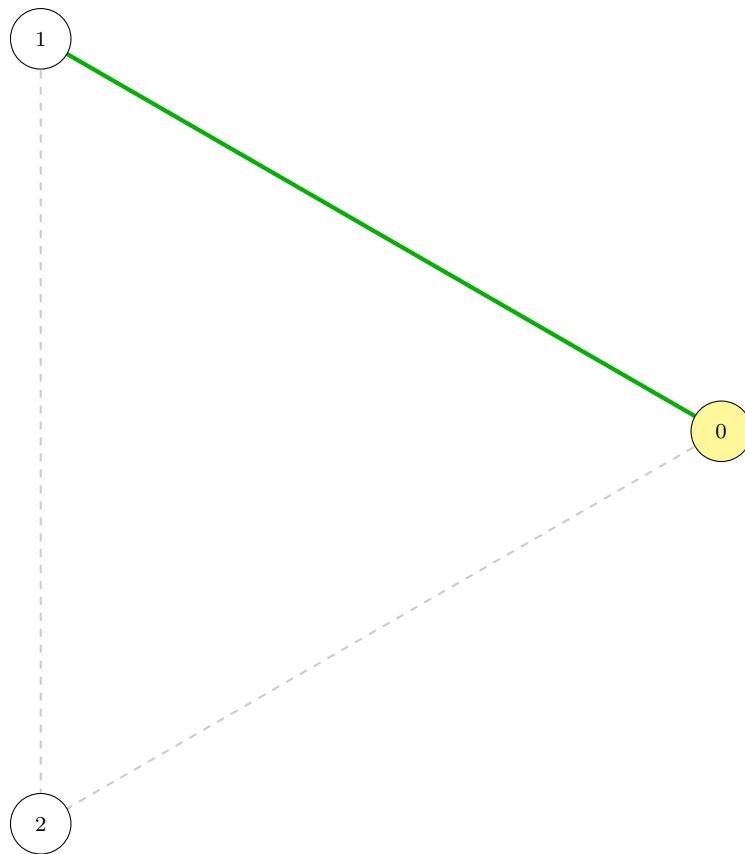
#### 11.0.1 Paso 1

Inicio del algoritmo. Se comienza desde el vértice 0.



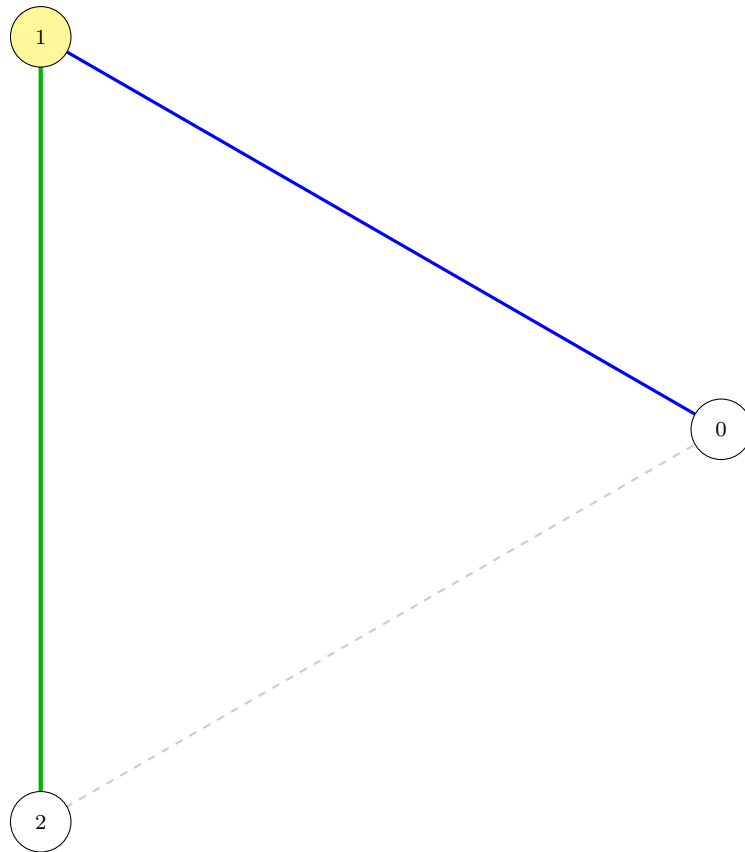
#### 11.0.2 Paso 2

Se elige la arista  $0 \rightarrow 1$ . Esta arista NO es un puente, por lo que es segura eliminarla sin desconectar el grafo.



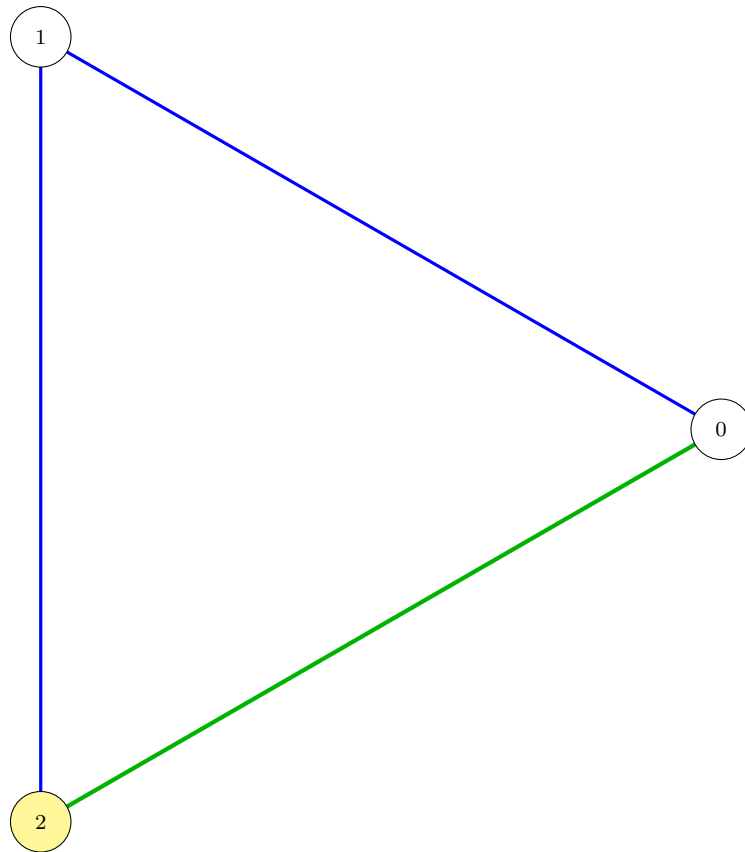
### 11.0.3 Paso 3

Se elige la arista  $1 \rightarrow 2$ . Esta arista NO es un puente, por lo que es segura eliminarla sin desconectar el grafo.



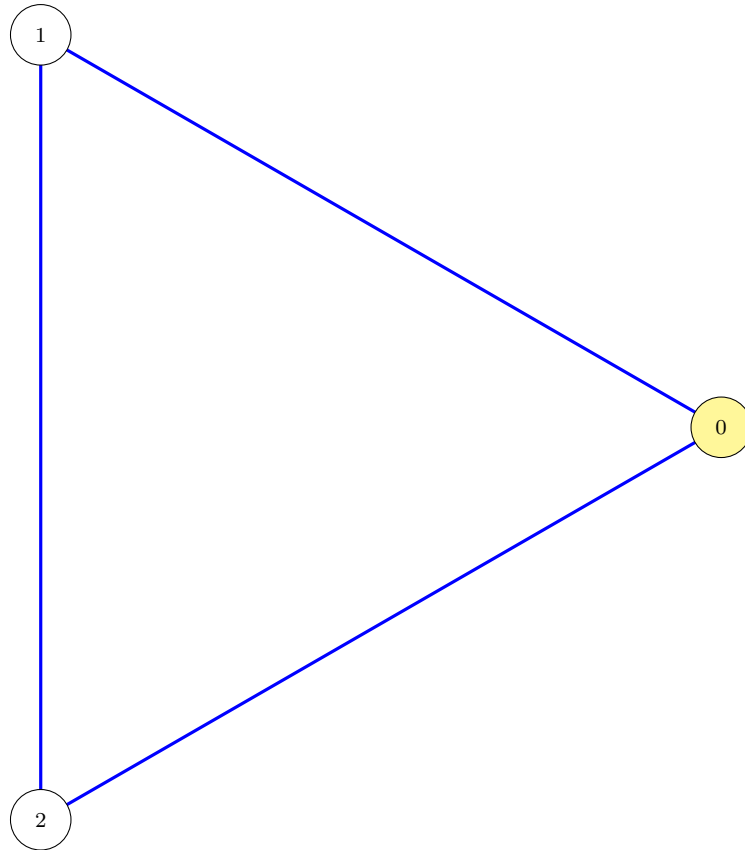
#### 11.0.4 Paso 4

Se elige la arista  $2 \rightarrow 0$ . Esta arista NO es un puente, por lo que es segura eliminarla sin desconectar el grafo.



#### 11.0.5 Paso 5

Finalización del algoritmo. Todas las aristas han sido eliminadas. Se ha construido un ciclo euleriano completo que regresa al vértice inicial 0.



#### Ciclo Euleriano Final:

El algoritmo encontró el siguiente ciclo euleriano completo:

$$0 \rightarrow 1 \rightarrow 2 \rightarrow 0$$

#### Análisis de la Ejecución:

El ciclo encontrado tiene 4 vértices. En cada paso, el algoritmo seleccionó una arista evitando puentes cuando era posible, garantizando que siempre quede un camino para regresar al vértice inicial. Al final, todas las aristas fueron eliminadas y se formó un ciclo euleriano completo.

**Complejidad:** El algoritmo de Fleury tiene complejidad temporal  $O(m^2)$  en el peor caso, donde  $m$  es el número de aristas, debido a la necesidad de verificar si una arista es un puente en cada paso.

## 12 Trabajo Extra Opcional: Ejecución Paso a Paso del Algoritmo de Fleury para Rutas Eulerianas

**Este grafo no es semieuleriano, por lo que no contiene una ruta euleriana.**

Para que un grafo sea semieuleriano y contenga una ruta euleriana, debe cumplir las siguientes condiciones:

1. El grafo debe ser conexo
2. Debe tener exactamente dos vértices con grado impar



Como este grafo no cumple estas condiciones, no es posible encontrar una ruta euleriana mediante el algoritmo de Fleury. Por lo tanto, no se puede mostrar la ejecución paso a paso del algoritmo para este caso.