Chinese Postman

Graph Theory and Social Networks

Contents

1. Introduction
2. Theoretical Background
3. Algorithm Overview
4. Implementation
5. Visualization of the Eulerian Circuit

1. Introduction

- The goal of the Chinese Postman Problem is to find the shortest closed walk that traverses every edge of a graph at least once.

- If the graph is Eulerian (all vertices have even degree), the solution is simply an Eulerian circuit.

- Most real-world graphs are non-Eulerian, so the problem is solved by duplicating selected edges to pair odd-degree vertices with minimum additional cost.

2. Theoretical background

- An **Eulerian circuit** is a closed walk that traverses every edge of a graph exactly once.
- A **connected graph** has an Eulerian circuit if and only if all vertices have **even degree**.
- If a graph contains **odd-degree vertices**, it is non-Eulerian and such a circuit does not exist.
- The Chinese Postman Problem addresses this by **adding edges** with **minimum total cost** to make all vertex degrees even.

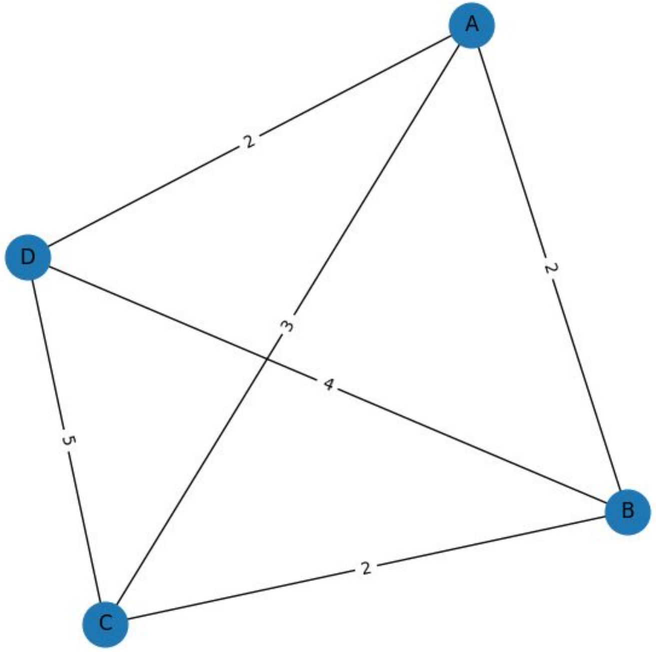
3. Algorithm overview

1. Check that the graph is connected.
2. Identify all vertices with odd degree.
3. Compute shortest paths between all odd-degree vertices.
4. Find a minimum-weight perfect matching among odd-degree vertices.
5. Duplicate the edges corresponding to the matched shortest paths.
6. Verify that the augmented graph is Eulerian.
7. Compute an Eulerian circuit in the augmented graph.

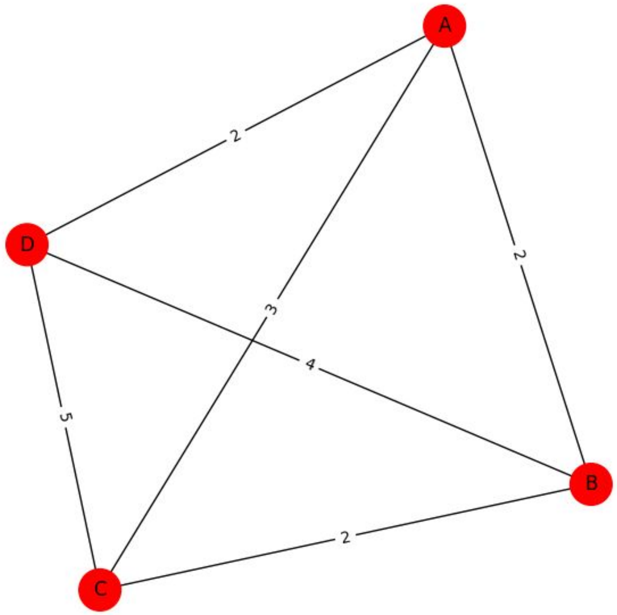
4. Implementation

- Connected graph that is not Eulerian and contains four vertices with odd degree.

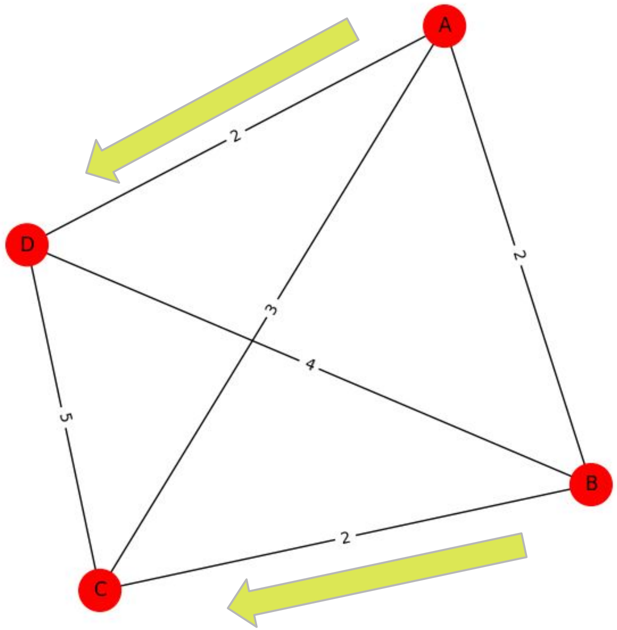
- Since the graph is non-Eulerian, an Eulerian circuit does not exist initially.



4.1 Detection of Odd-Degree Vertices

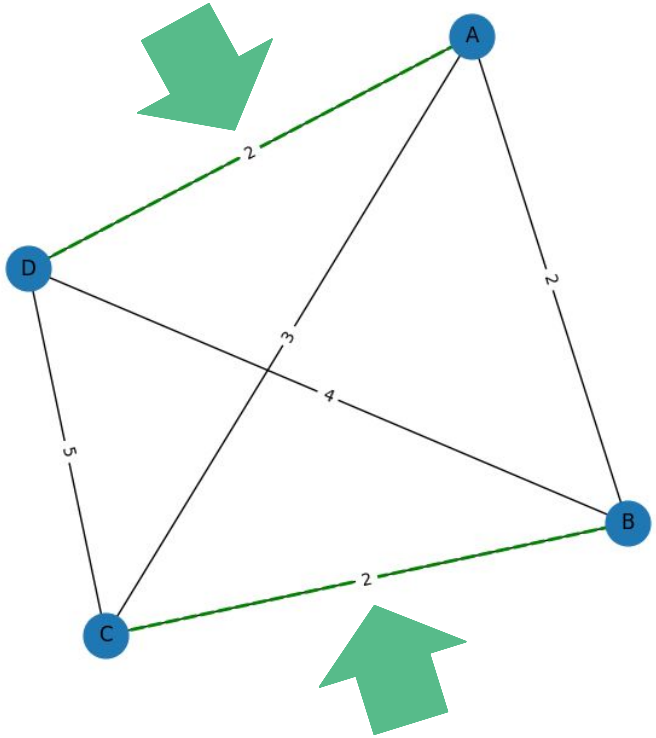


4.2 Shortest paths and Minimum Weight Matching



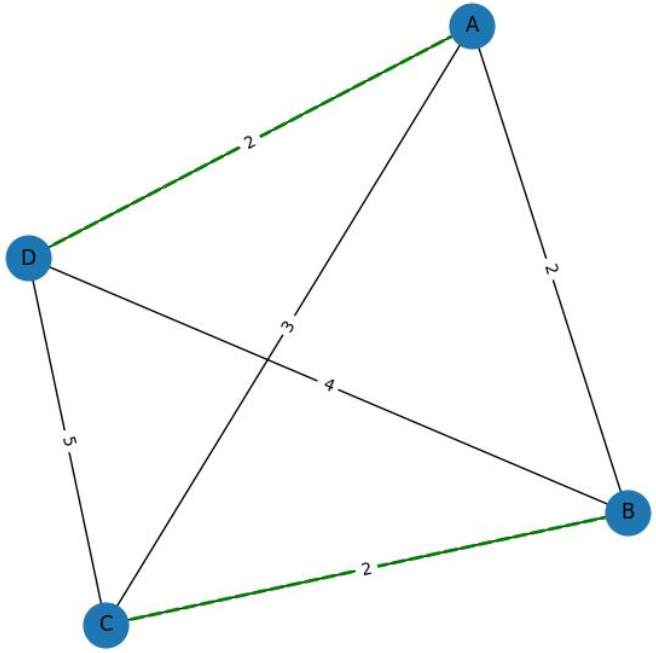
4.3 Graph Augmentation

These are duplicated edges added to the graph (to ensure every node has an even degree).

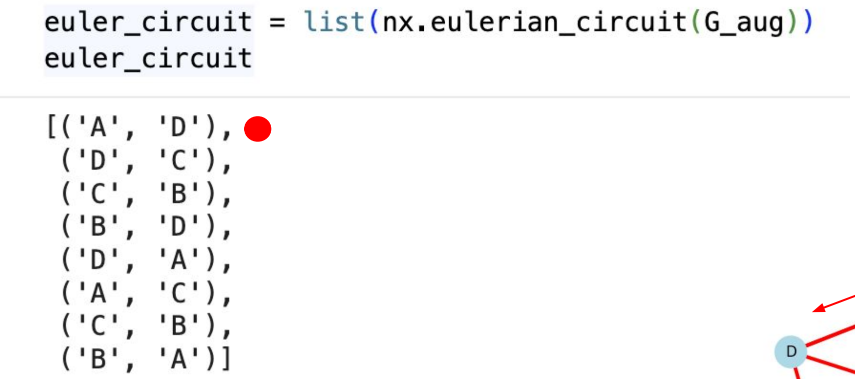


4.4 Verify that the augmented graph is Eulerian

As the graph is connected and all vertices have even degree, it is Eulerian (meaning a continuous path exists that traverses every edge exactly once)

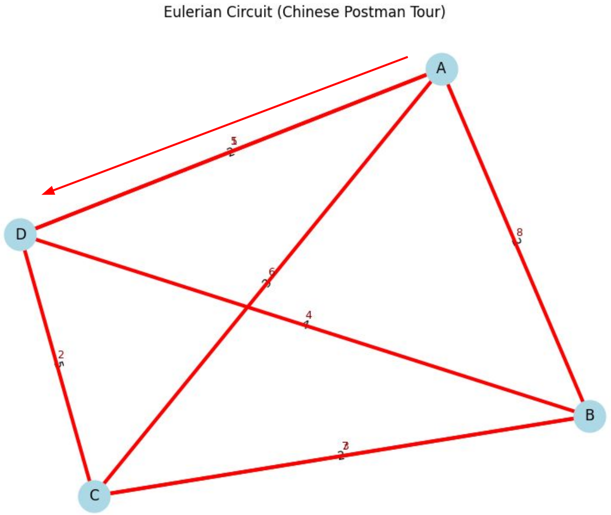


5. Visualization of the Eulerian Circuit

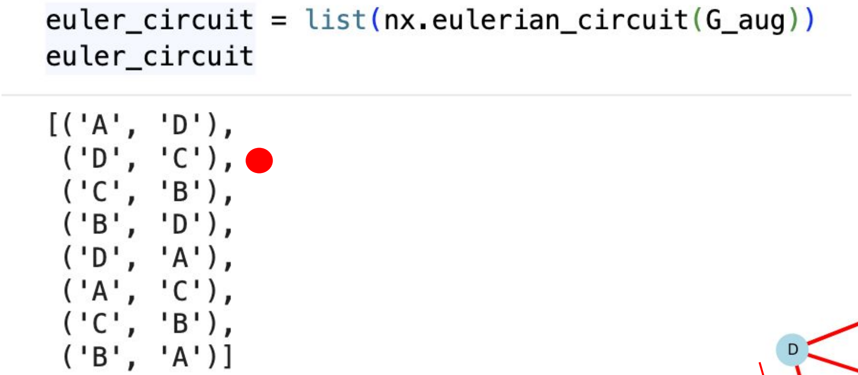


```
euler_circuit = list(nx.eulerian_circuit(G_aug))
euler_circuit
```

```
[('A', 'D'),
 ('D', 'C'),
 ('C', 'B'),
 ('B', 'D'),
 ('D', 'A'),
 ('A', 'C'),
 ('C', 'B'),
 ('B', 'A')]
```

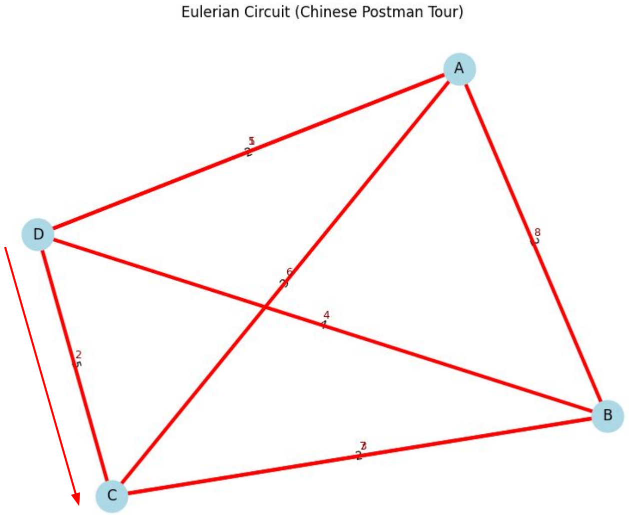


5. Visualization of the Eulerian Circuit



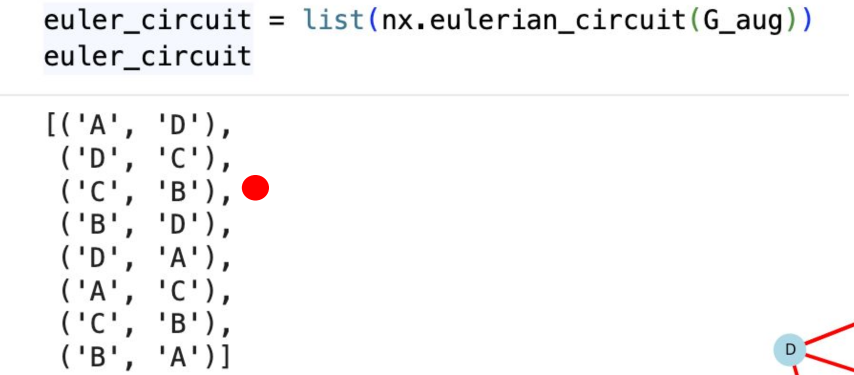
```
euler_circuit = list(nx.eulerian_circuit(G_aug))
euler_circuit
```

```
[('A', 'D'),
 ('D', 'C'), ●
 ('C', 'B'),
 ('B', 'D'),
 ('D', 'A'),
 ('A', 'C'),
 ('C', 'B'),
 ('B', 'A')]
```



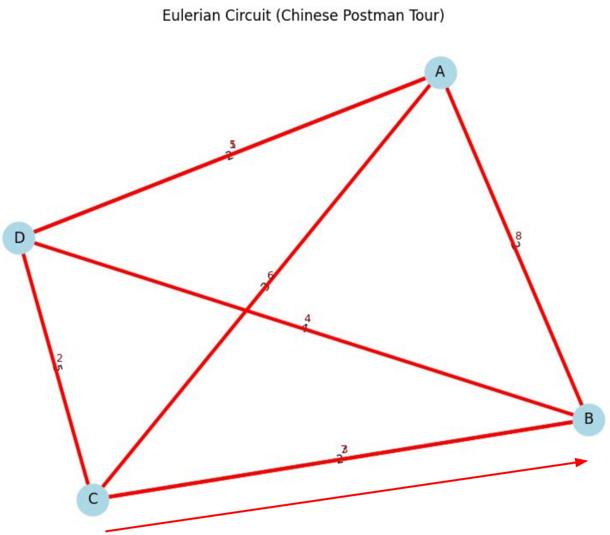
Eulerian Circuit (Chinese Postman Tour)

5. Visualization of the Eulerian Circuit



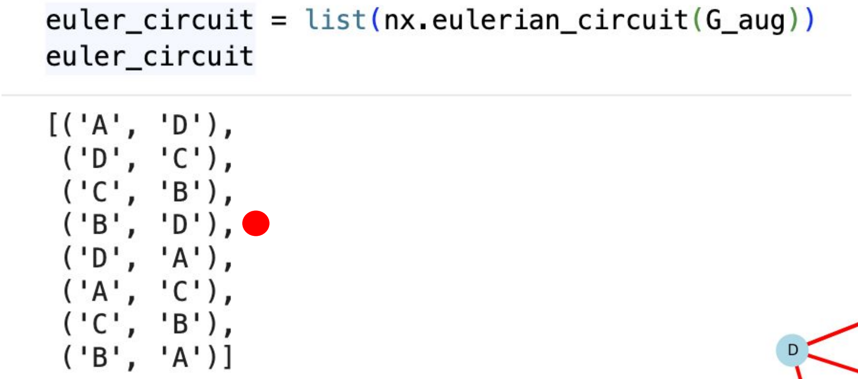
```
euler_circuit = list(nx.eulerian_circuit(G_aug))
euler_circuit
```

```
[('A', 'D'),
 ('D', 'C'),
 ('C', 'B'), ●
 ('B', 'D'),
 ('D', 'A'),
 ('A', 'C'),
 ('C', 'B'),
 ('B', 'A')]
```



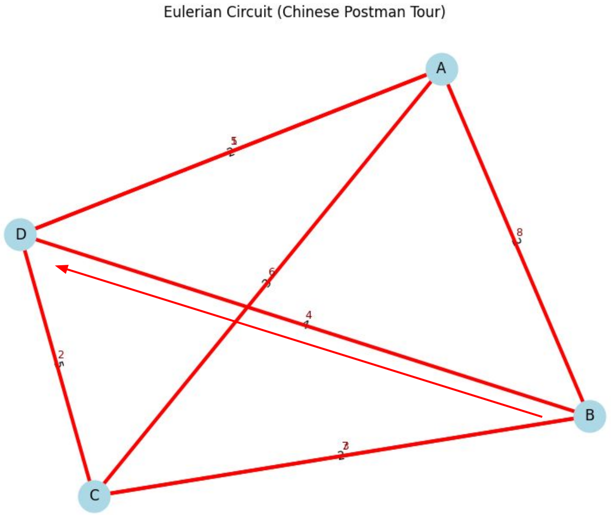
Eulerian Circuit (Chinese Postman Tour)

5. Visualization of the Eulerian Circuit



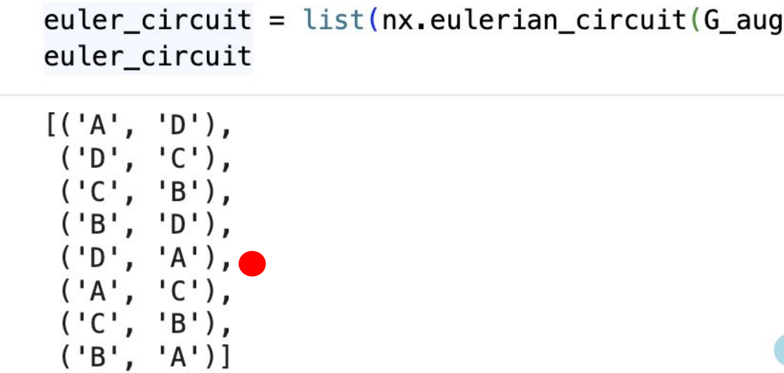
```
euler_circuit = list(nx.eulerian_circuit(G_aug))
euler_circuit
```

```
[('A', 'D'),
 ('D', 'C'),
 ('C', 'B'),
 ('B', 'D'), ●
 ('D', 'A'),
 ('A', 'C'),
 ('C', 'B'),
 ('B', 'A')]
```



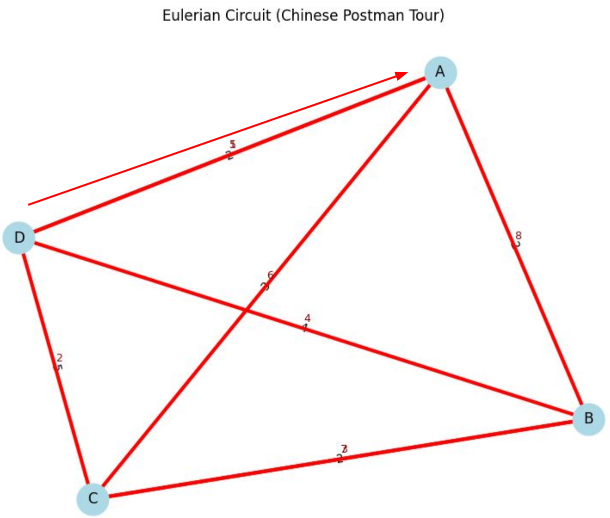
Eulerian Circuit (Chinese Postman Tour)

5. Visualization of the Eulerian Circuit

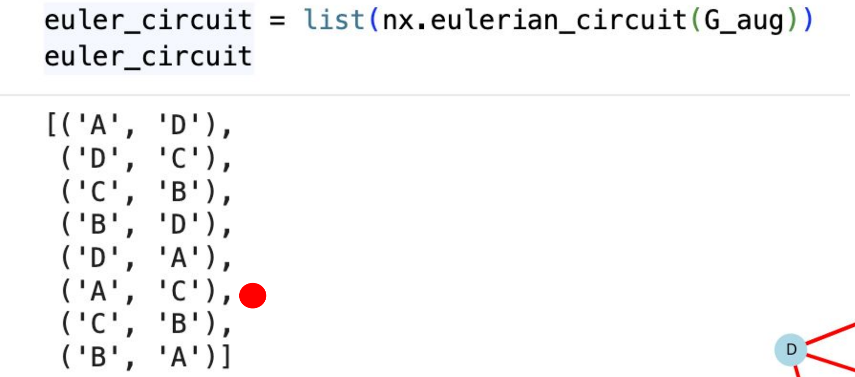


```
euler_circuit = list(nx.eulerian_circuit(G_aug))
euler_circuit
```

```
[('A', 'D'),
 ('D', 'C'),
 ('C', 'B'),
 ('B', 'D'),
 ('D', 'A'),
 ('A', 'C'),
 ('C', 'B'),
 ('B', 'A')]
```



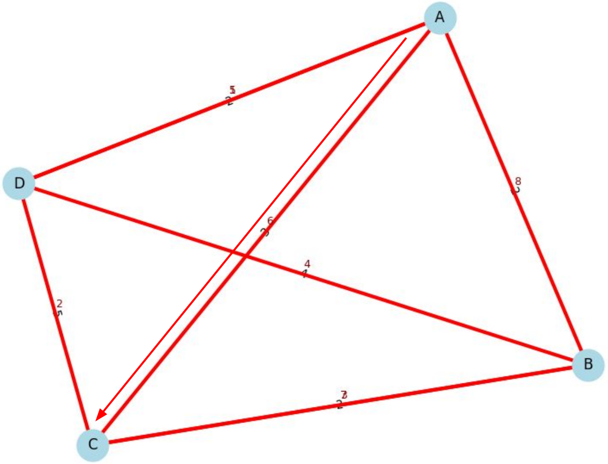
5. Visualization of the Eulerian Circuit



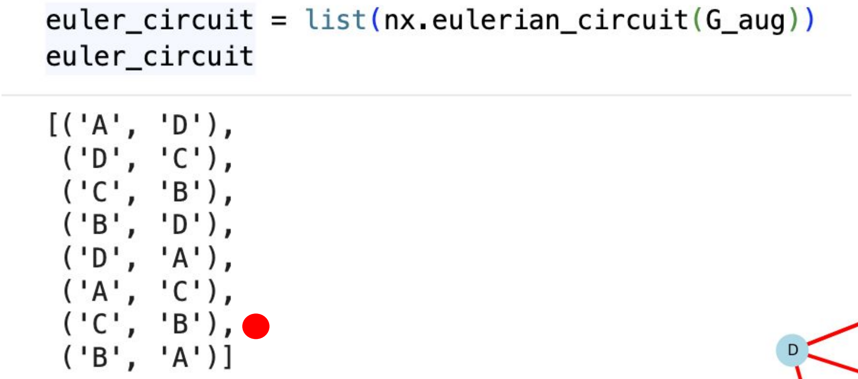
```
euler_circuit = list(nx.eulerian_circuit(G_aug))
euler_circuit
```

```
[('A', 'D'),
 ('D', 'C'),
 ('C', 'B'),
 ('B', 'D'),
 ('D', 'A'),
 ('A', 'C'), ●
 ('C', 'B'),
 ('B', 'A')]
```

Eulerian Circuit (Chinese Postman Tour)

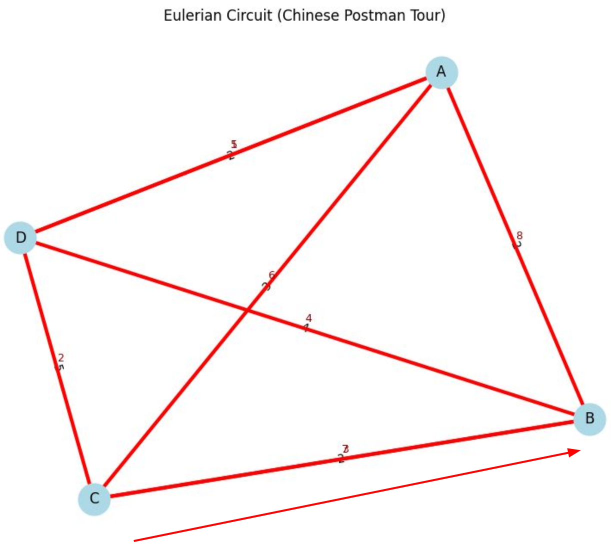


5. Visualization of the Eulerian Circuit

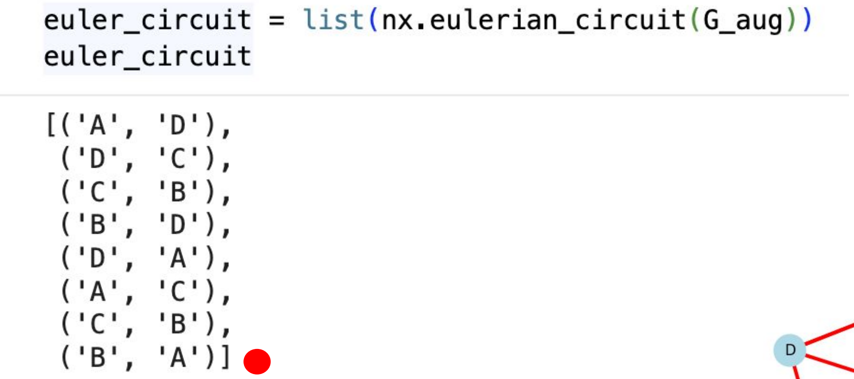


```
euler_circuit = list(nx.eulerian_circuit(G_aug))
euler_circuit
```

```
[('A', 'D'),
 ('D', 'C'),
 ('C', 'B'),
 ('B', 'D'),
 ('D', 'A'),
 ('A', 'C'),
 ('C', 'B'), ●
 ('B', 'A')]
```

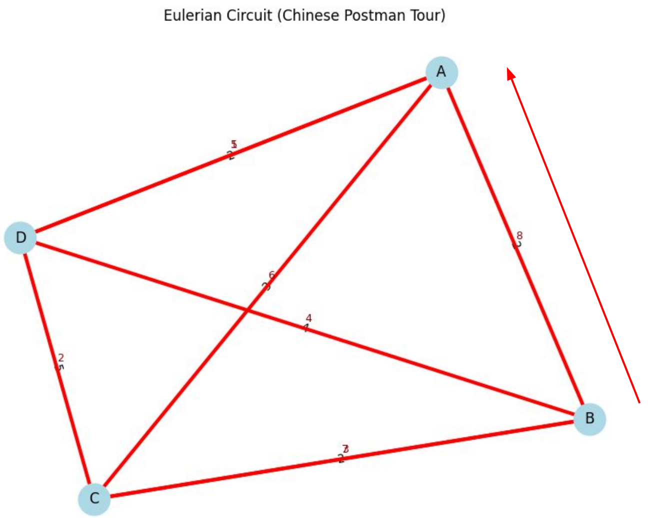


5. Visualization of the Eulerian Circuit



```
euler_circuit = list(nx.eulerian_circuit(G_aug))
euler_circuit
```

```
[('A', 'D'),
 ('D', 'C'),
 ('C', 'B'),
 ('B', 'D'),
 ('D', 'A'),
 ('A', 'C'),
 ('C', 'B'),
 ('B', 'A')]
```



Eulerian Circuit (Chinese Postman Tour)

Total cost of the Tour is 22