

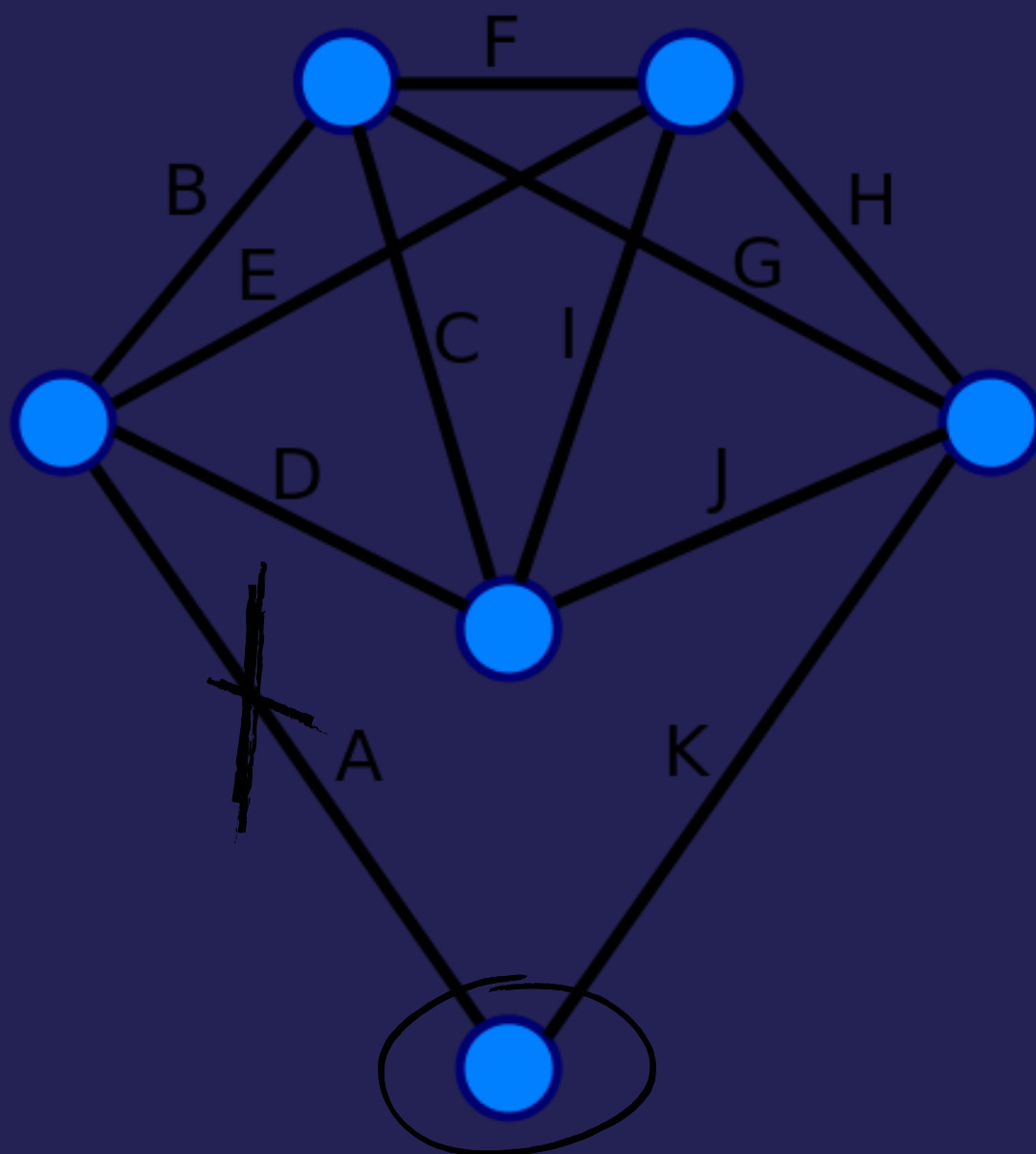
# Algoritmo de Fleury

Joel Victor, Raynan Serafim



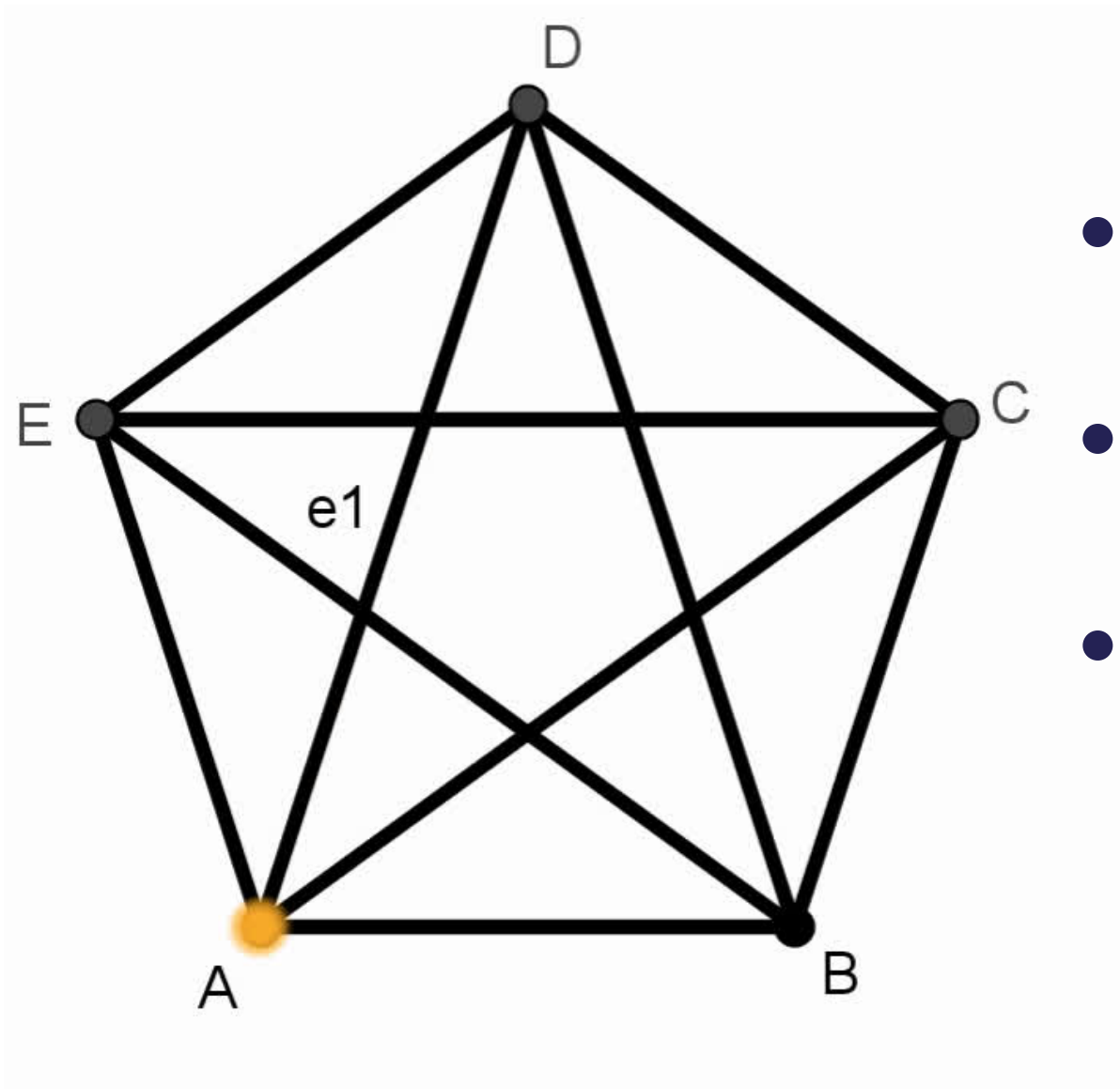
INSTITUTO  
FEDERAL  
Ceará

# Trilha e Ciclo(Circuito) Euleriano(a)



- Presentes em grafos não dirigidos e conexos
- Trilha é o caminho em que todos os vértices do grafo são visitadas
- Uma trilha que possui um ciclo é chamada de ciclo Euleriano

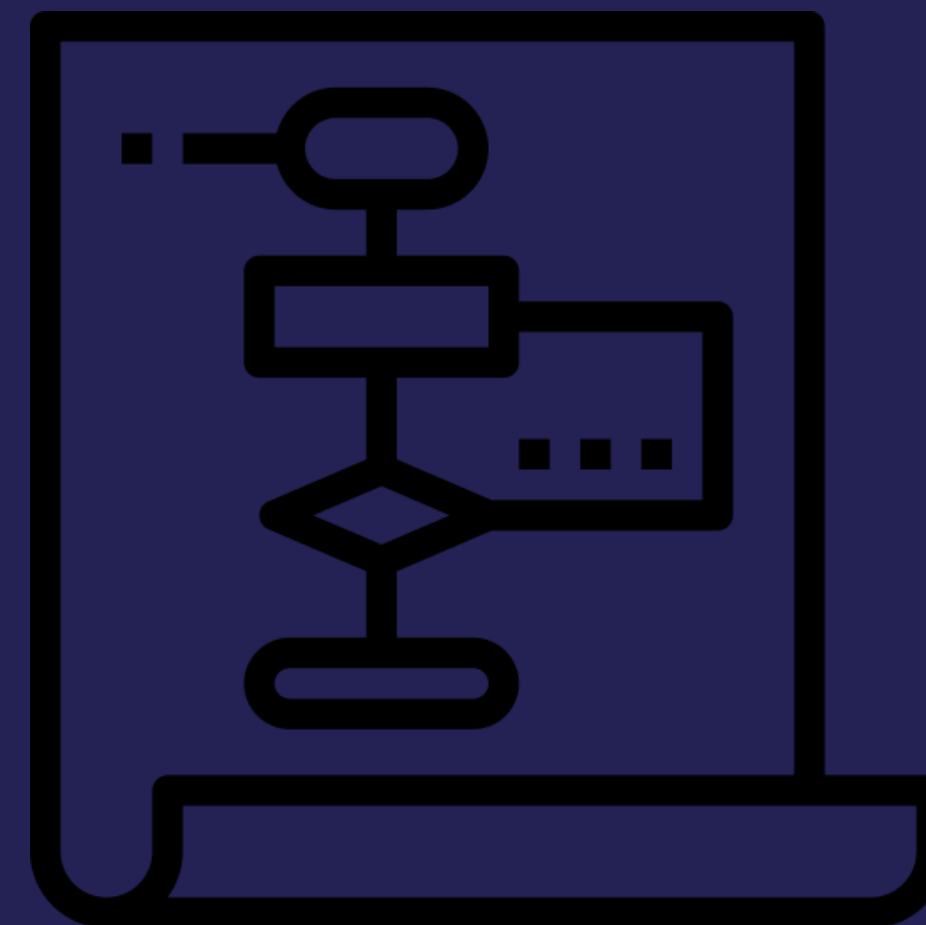
# Trilha e Ciclo(Circuito) Euleriano(a)



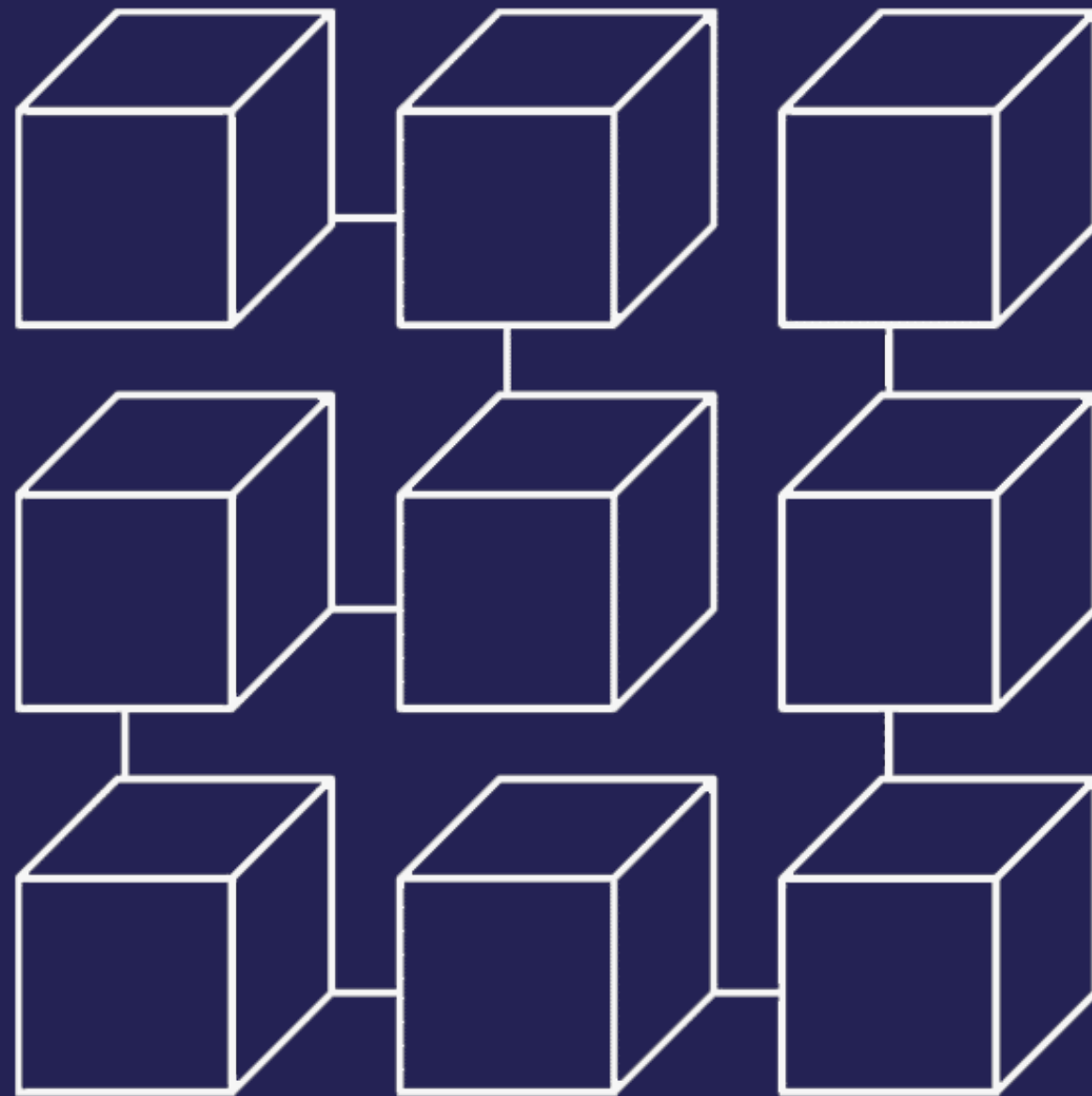
- Para a formação de uma trilha, é preciso que o número de vértices com grau ímpar, seja 0 ou 2
- Caso o grafo possua 2 vértices de grau ímpar, o inicia em uma delas e termina na outra
- Caso o grafo não possua vértices ímpares, é possível criar um ciclo Euleriano a partir de qualquer vértice

# Algoritmo de Fleury

- É um algoritmo elegante e simples que data de 1883.
- É usado para descobrir se um grafo possui as características de um grafo Euleriano e produzir trilhas ou ciclos possíveis.

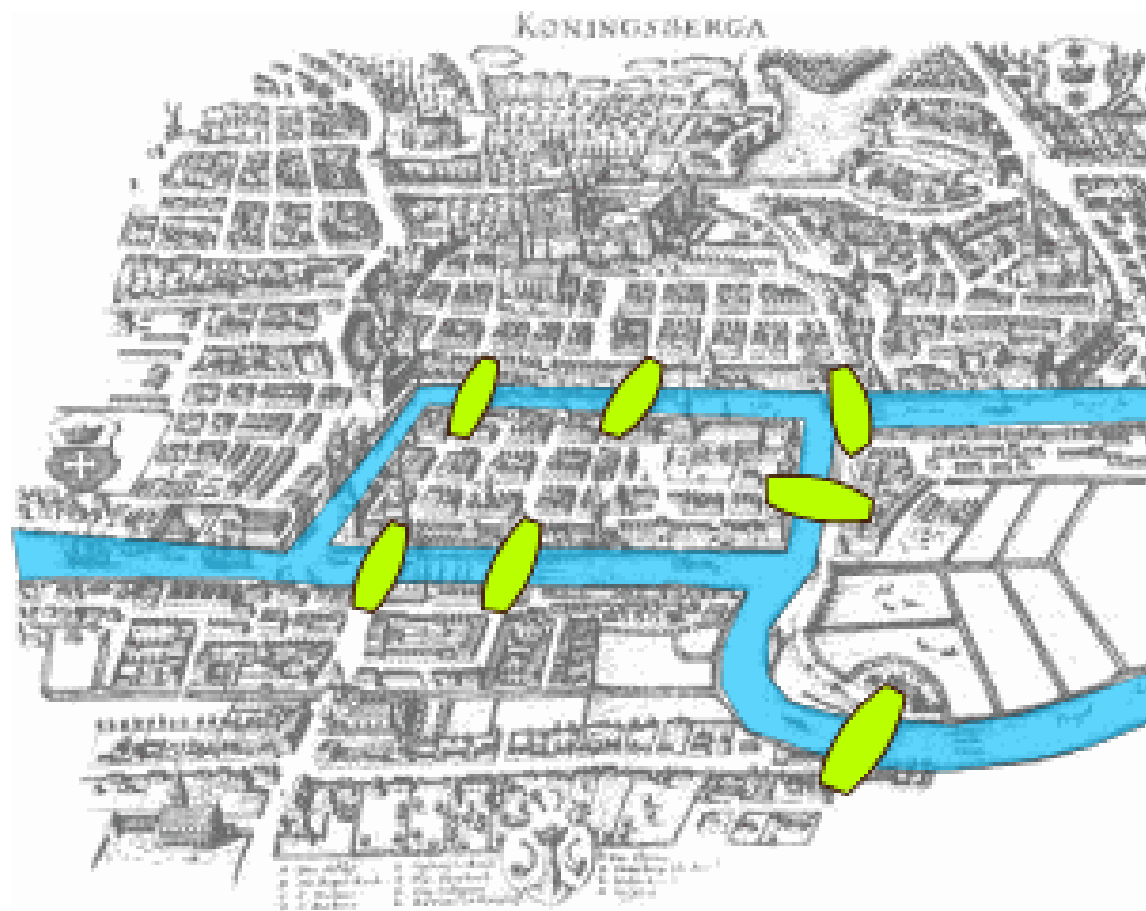


# Estrutura de dados

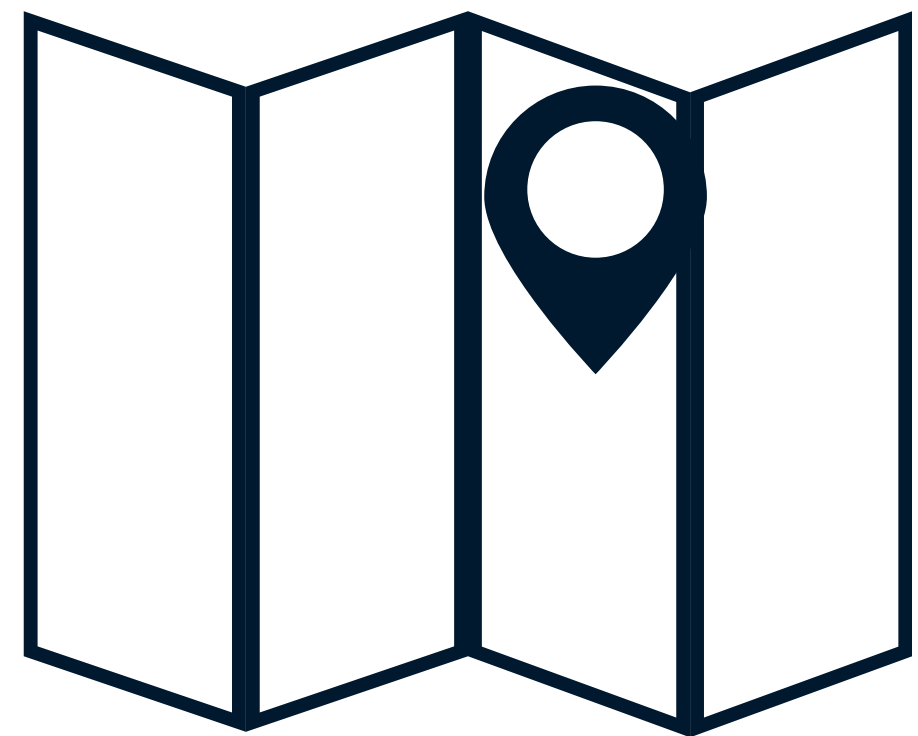
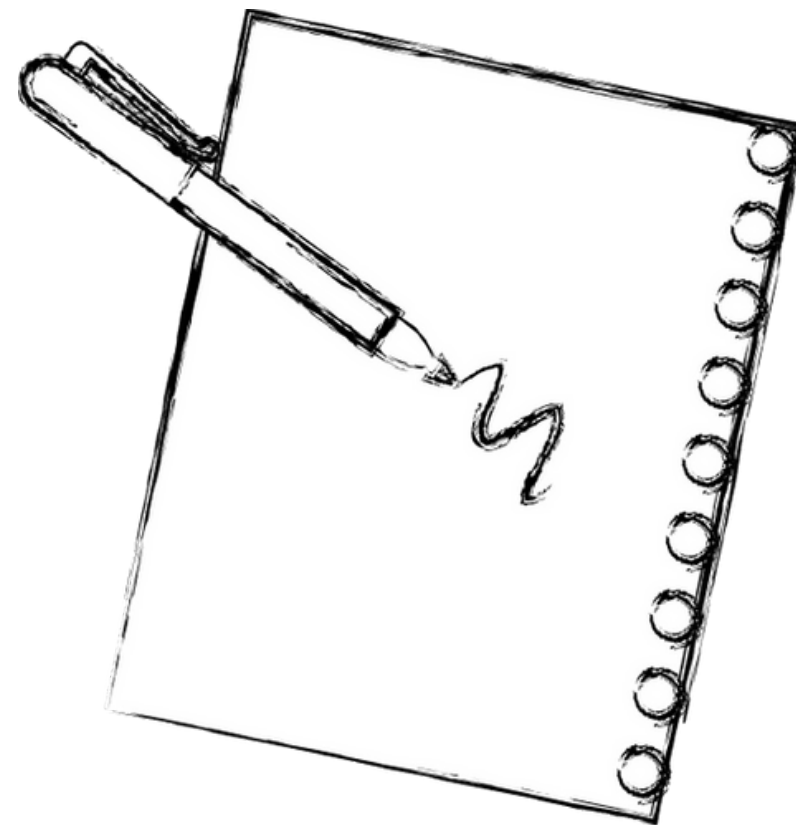


- Matriz de adjacência
- Estrutura de pilha para o algoritmo de DFS
- Variaveis diversas

# Aplicações

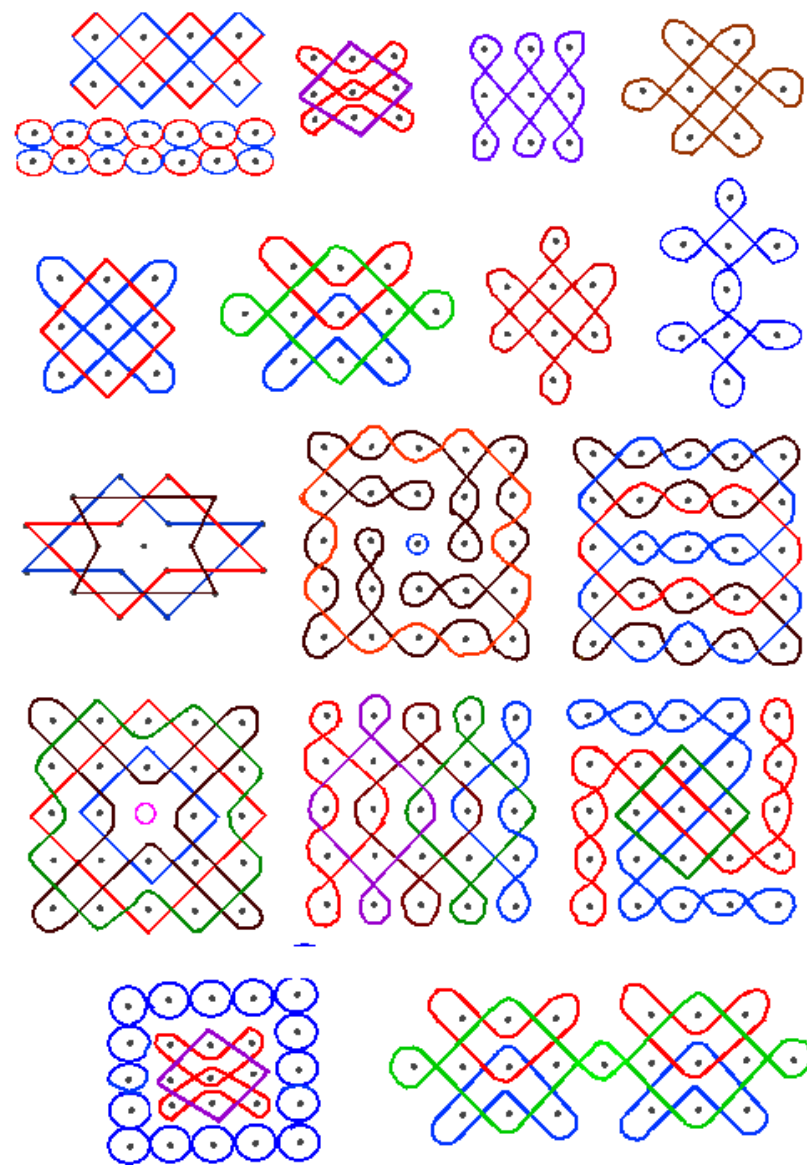


- Analise de problemas classicos
  - Problema das 7 pontes de Königsberg
  - Traçar diagrama de pontos em uma folha
  - Problemas de inspeção de rota

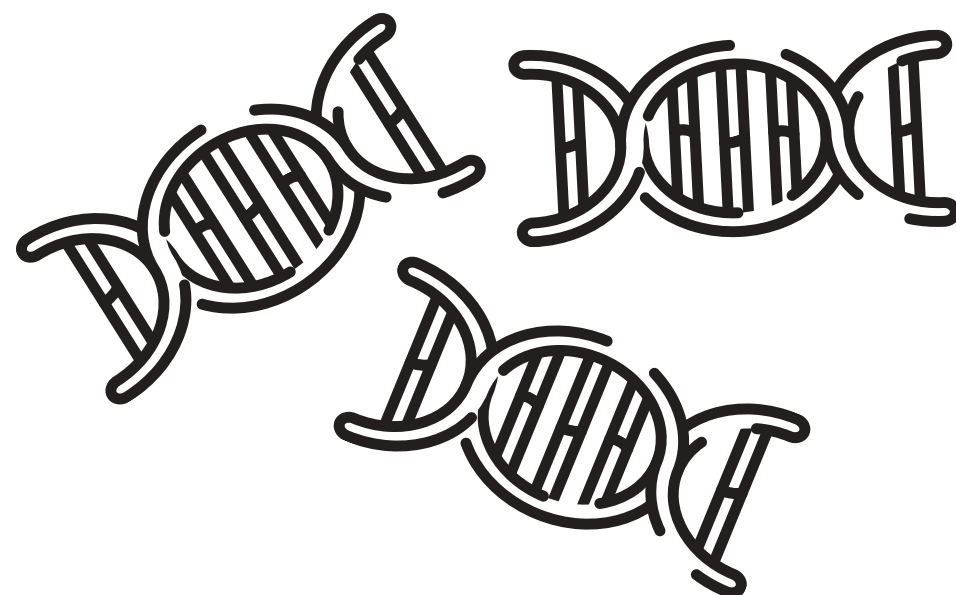




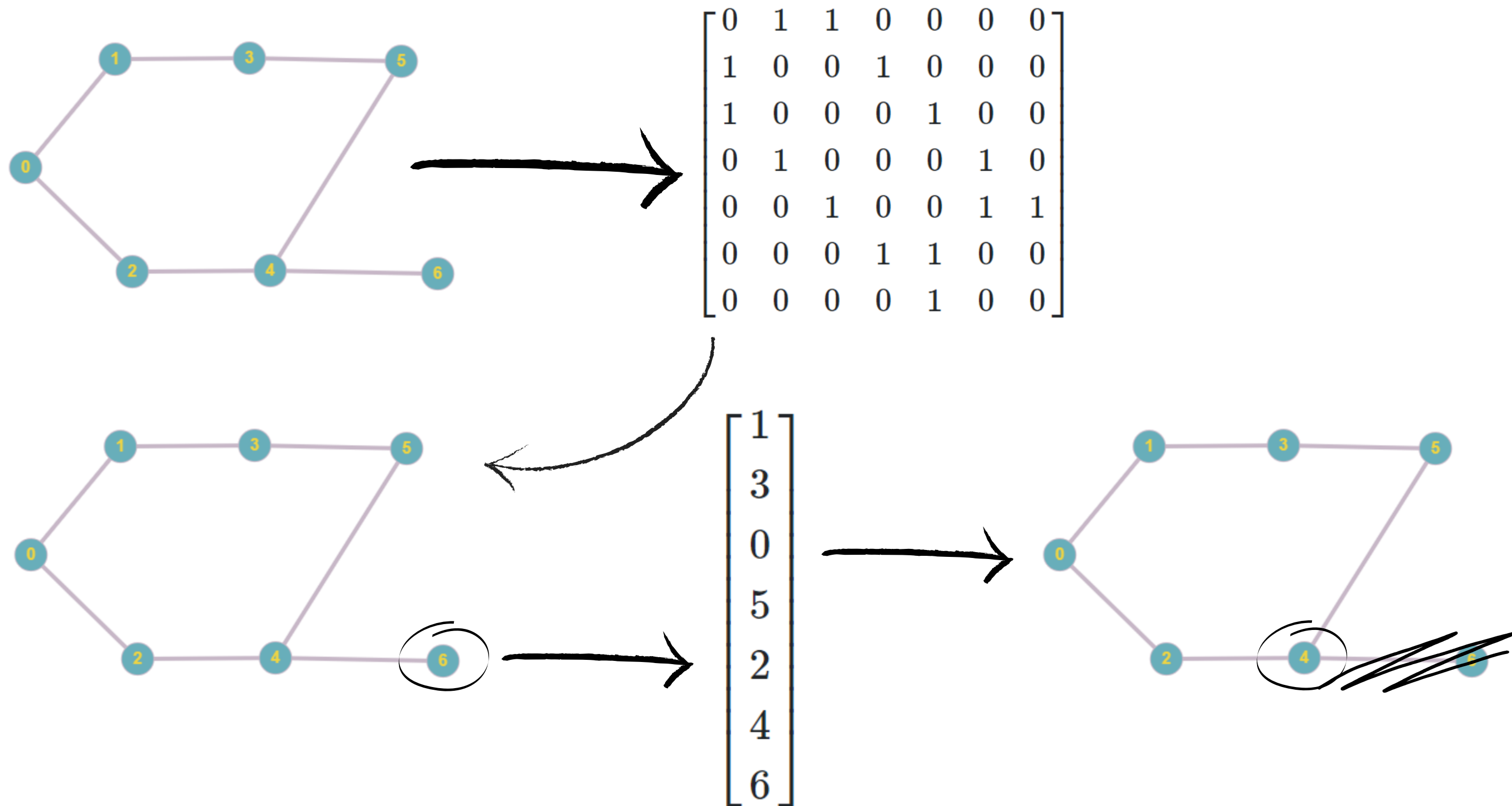
# Aplicações



- Anexando informações a cada vertice, é possível resolver alguns tipos de problemas que podem ser representados com grafos eulerianos
  - Reconstrução de fragmentos de DNA na área de Biologia Computacional
  - Geração de rotas para entregas em uma representação em grafos de uma cidade
  - Geração de padrões de desenho Kolam

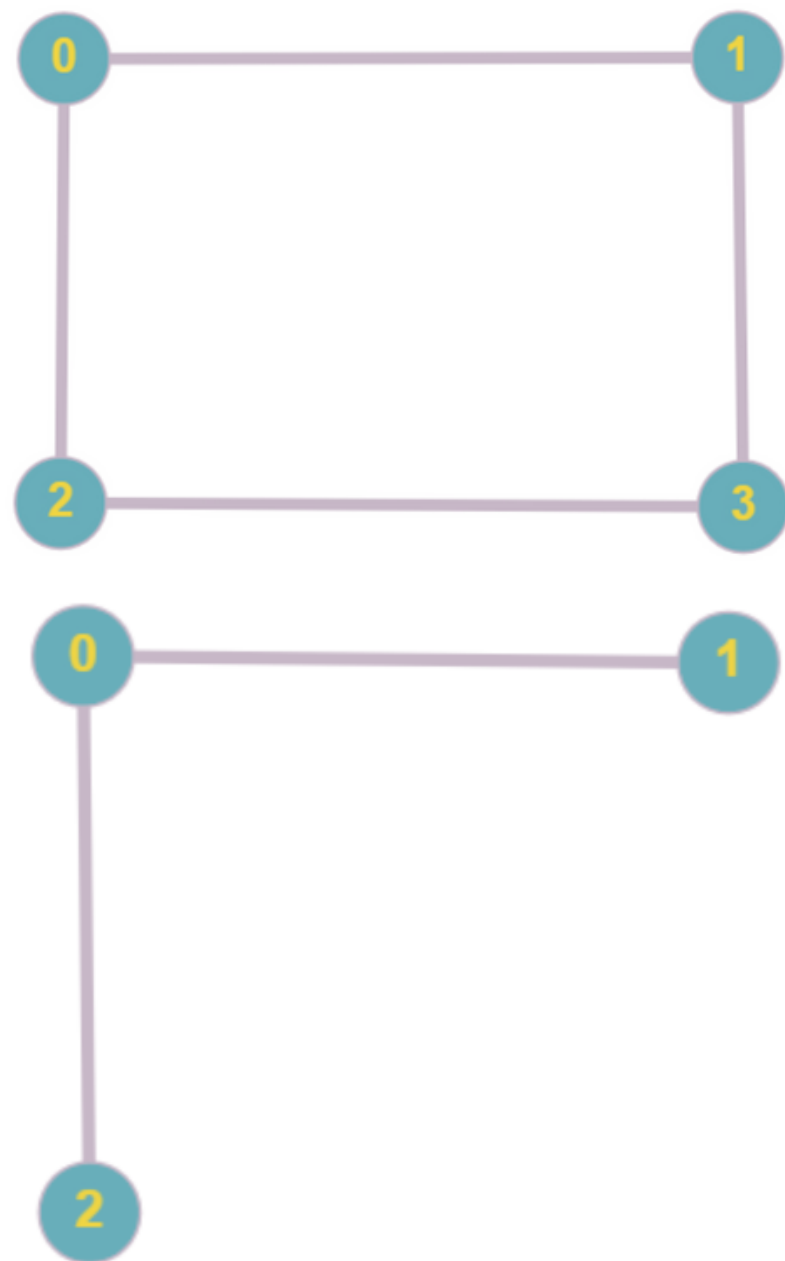


# Algoritmo





# Testes e validação

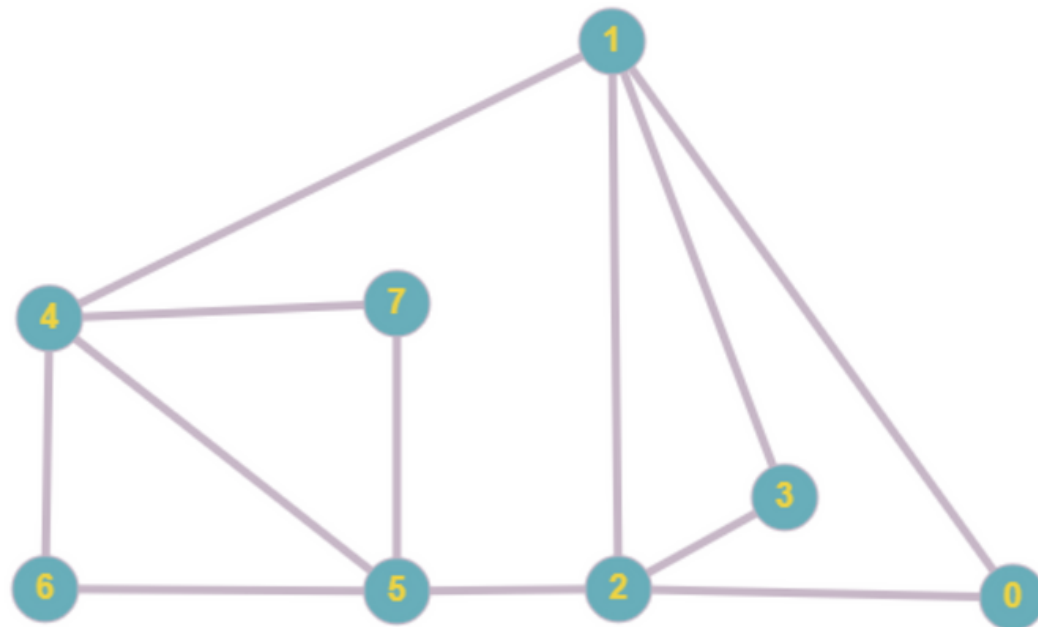


Graph Online:  $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 0$   
Tutorials Point:  $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 0$   
Geeks for Geeks:  $0 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 0$   
Autoria:  $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 0$

Graph Online:  $2 \rightarrow 0 \rightarrow 1$   
Tutorials Point:  $1 \rightarrow 0 \rightarrow 2$   
Geeks for Geeks:  $1 \rightarrow 0 \rightarrow 2$   
Autoria:  $1 \rightarrow 0 \rightarrow 2$



# Testes e validação

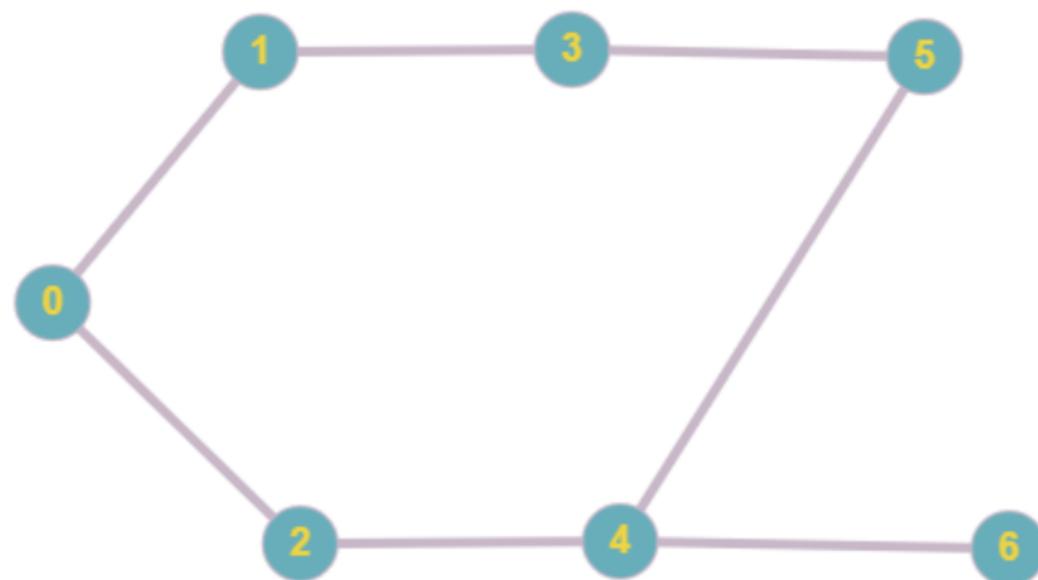


G.O: 0->1->3->2->5->6->4->7->5->4->1->2->0

T.P: 0->1->2->3->1->4->5->6->4->7->5->2->0

G.G: 0->1->3->2->1->4->6->5->4->7->5->2->0

Autoria: 0->1->2->3->1->4->5->6->4->7->5->2->0

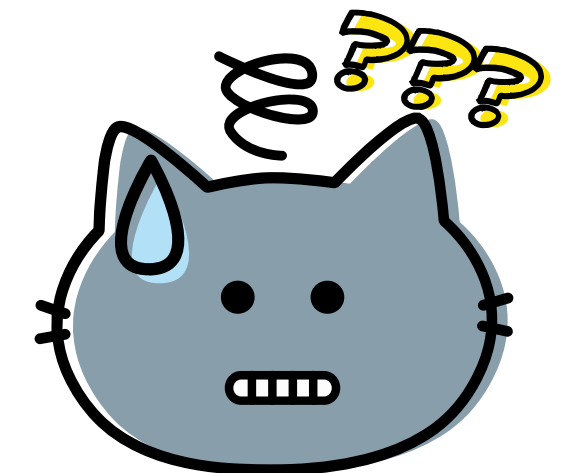


Graph Online: 6 -> 4 -> 2 -> 0 -> 1 -> 3 -> 5 -> 4

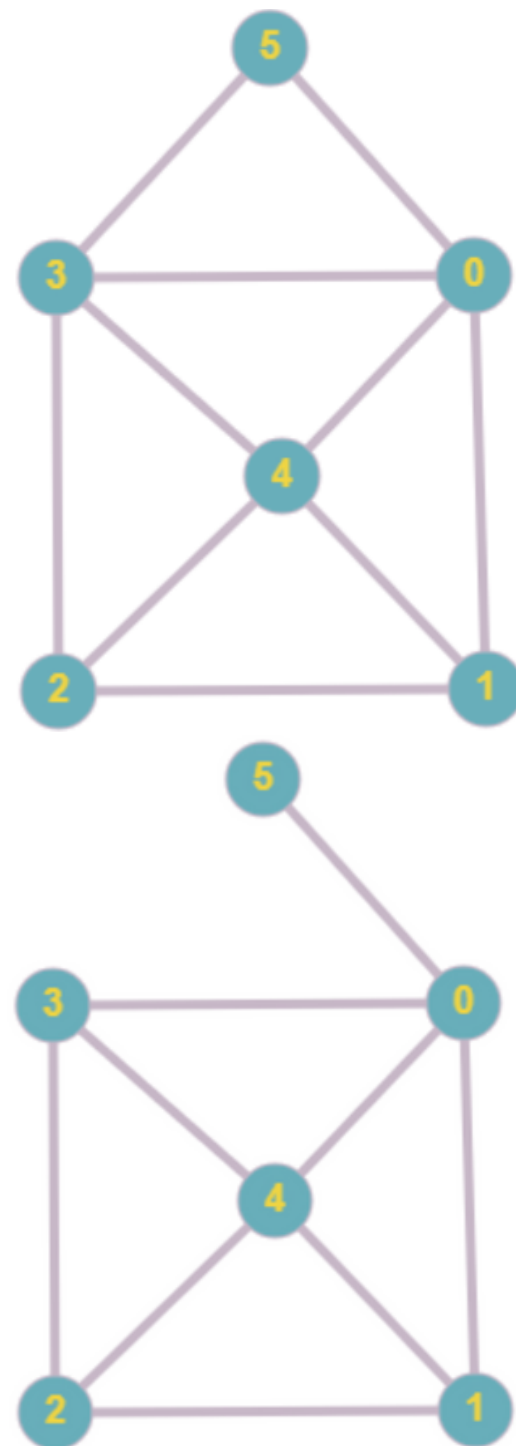
Tutorials Point: 4 -> 2 -> 0 -> 1 -> 3 -> 5 -> 4 -> 6

Geeks for Geeks: 4 -> 2 -> 0 -> 1 -> 3 -> 5 -> 4 -> 6

Autoria: 4 -> 2 -> 0 -> 1 -> 3 -> 5 -> 4 -> 6

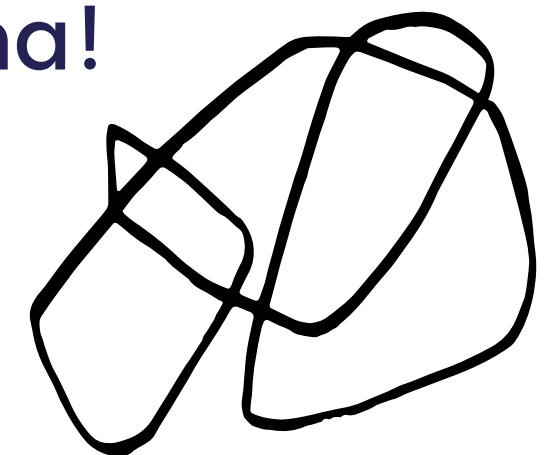


# Testes e validação



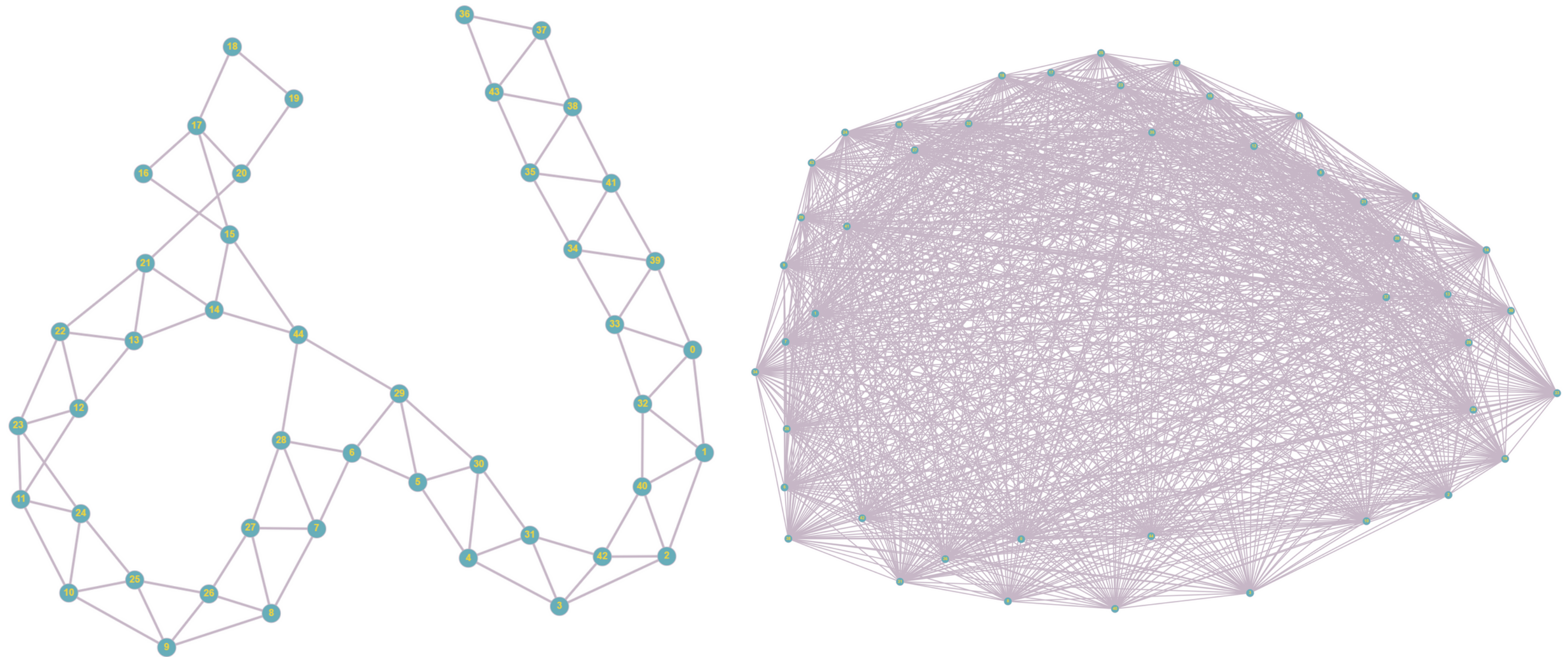
G.O: 2→1→0→3→2→4→0→5→3→4→1  
T.P: 1→0→3→2→1→4→0→5→3→4→2  
G.G: 1→0→3→2→1→4→3→5→0→4→2  
Autoria: 1→4→3→5→0→1→2→3→0→4→2

Graph Online: Graph has not Eulerian path  
T.P: 1→0→3→2→1→4→3 ???? 2→4→0→5  
Geeks for Geeks: Loop infinito!  
Autoria: O grafo não possui trilha euleriana!





# Testes e validação



# Testes e validação

[illegible]

# Dificuldades



- Iniciais
  - Conhecimento em teoria de grafos
- Implementação
  - Desenvolver um método de validação para aresta de corte
  - Trabalhar DFS sem recursividade
- Bugs/erros
  - Validar saída para diversos casos



# Bibliografia

- <https://www.inf.ufsc.br/grafos/temas/euleriano/euleriano.htm>
- <http://www.gpec.ucdb.br/pistori/disciplinas/discreta/aulas/geulham.htm>
- <https://www.youtube.com/watch?v=mRvt5yJT7E>
- <http://tmrfindia.org/sutra/v2i21.pdf>
- <https://www.ic.unicamp.br/~meidanis/research/fa/>
- <https://www.geeksforgeeks.org/eulerian-path-and-circuit/>
- <https://www.tutorialspoint.com/fleury-s-algorithm-for-printing-eulerian-path-or-circuit-in-cplusplus>
- <https://www.tutorialspoint.com/Fleury-s-Algorithm>
- <https://www.geeksforgeeks.org/implementation-of-bfs-using-adjacency-matrix/>



**THANK YOU!**