

# Relatório da Atividade 1 de Grafos

Disciplina: INE5413-04208 (20241) - Grafos

Professores: Rafael Santiago

Aluno: Lucas Tinoco e Enzo Welter

Data: 28/04/2024

---

## 1. Representação

No código da classe Grafo foram escolhidas listas para armazenar os vértices e as arestas do grafo, pois são estruturas simples e eficientes para manipular esses elementos. Além disso, foi utilizada um dicionário para armazenar os pesos das arestas. Essas estruturas foram escolhidas para representar grafos de forma eficiente e facilitar operações como busca, verificação de adjacência e cálculo de pesos.

## 2. Buscas

No código, foram escolhidas listas para os conjuntos de vértices conhecidos, distâncias e ancestrais, devido à sua simplicidade e eficiência. A fila (Queue) foi utilizada para a ordem de visita dos vértices na BFS, enquanto o DataFrame do Pandas foi adotado para exibir os resultados de forma tabular, facilitando a análise. Essas estruturas de dados foram selecionadas visando a implementação eficiente do algoritmo BFS e a organização clara dos resultados.

## 3. Ciclo Euleriano

O algoritmo utiliza um dicionário para marcar as arestas como visitadas ou não. O código também utiliza listas para armazenar os vértices do ciclo euleriano encontrado e os vértices abertos que ainda precisam ser explorados. Essas escolhas facilitam a implementação do algoritmo de identificação de ciclo

euleriano e para a manipulação dos vértices e arestas do grafo, além de serem eficientes.

## 4. Algoritmo de Dijkstra

O código utiliza dicionários para armazenar distâncias (D), ancestrais (A), e para marcar os vértices como visitados ou não (C). Além disso, listas são utilizadas para armazenar o caminho mínimo para cada vértice destino. O algoritmo de Dijkstra é então executado, e o resultado é formatado conforme especificado, com o destino, o caminho percorrido e a distância necessária, antes de ser impresso na tela. Essas estruturas de dados também facilitam a implementação do algoritmo de Dijkstra e permitem organizar o resultado de forma clara e eficiente.

## 5. Algoritmo de Floyd-Warshall

O código utiliza uma matriz de distâncias para armazenar as distâncias mínimas entre todos os pares de vértices. A matriz é inicializada com infinito para representar distâncias desconhecidas. Em seguida, as distâncias conhecidas são preenchidas na matriz com base nas arestas do grafo. O algoritmo de Floyd-Warshall é então aplicado para calcular as distâncias mínimas entre todos os pares de vértices. Por fim, o resultado é impresso na tela conforme o formato especificado, mostrando as distâncias para cada vértice na ordem crescente dos índices informados no arquivo de entrada. Essa estrutura de matriz é ideal para armazenar e calcular as distâncias entre todos os pares de vértices de forma eficiente.