DIN Departamento de Informática

**Universidade Estadual de Maringá**
**Centro de Tecnologia**
**Departamento de Informática**

UEM Universidade Estadual de Maringá

# Trabalho de Algoritmos em Grafos.

## Implementação dos algoritmos de Prim e de Kruskal

Aluno : Guilherme José Monteiro Meirelles

Ra: 133994

Maringá
2025

# 1.Introdução

Este documento tem como intuito abordar sobre a implementação dos algoritmos de Prim e de Kruskal, a partir da contextualização sobre o tema e do relato dos resultados obtidos.

# 2. Fundamentação teórica

Grafos é um conjunto de objetos representados na forma de vértices e arestas que se conectam. Os vértices representam entidades e as arestas a associação destas entidades. Neste sentido, os grafos são muito utilizados para representar e resolver problemas matemáticos e do cotidiano.

Um problema específico é encontrar a árvore geradora mínima de um grafo, que é basicamente, o menor custo de conexões possíveis para que todos os vértices de um grafo estejam conectados, ou seja, o menor custo ou peso de arestas para que haja um caminho entre todos os vértices do grafo a partir das arestas.

Este problema pode aparecer no cotidiano, por exemplo: um eletricista necessita calcular o menor custo de conexões elétricas para que todos os aparelhos de um edifício sejam abrangidos. Neste sentido, dois algoritmos matemáticos responsáveis por encontrar a árvore geradora mínima de um grafo, são o algoritmo de Prim e o de Kruskal.

O algoritmo de Prim para encontrar a árvore geradora seleciona um vértice inicial e a partir disso ‘visita’ os vértices conectados com o menor custo, adicionando as arestas de menor custo de um vértice não visitado na árvore.

Já o algoritmo de Kruskal, ordena as arestas do grafo em ordem crescente de custo e adiciona sequencialmente as arestas de menor custo na árvore a não ser que ela gere um ciclo.

A partir disso desenvolveu-se um algoritmo em python para simular um grafo e executar os algoritmos de Prim e de Kruskal.

## 3. Resultados obtidos

Executando os algoritmos de Prim e de Kruskal no programa implementado. Notou-se que o algoritmo de Kruskal está sendo executado de forma mais eficiente, sendo mais rápido e tendo valores menores de pico de alocação de memória.

Provável hipótese sobre esta diferença é que o algoritmo de Prim implementado pelo estudante utiliza mais estruturas de dados auxiliares durante sua execução. Por exemplo: o algoritmo usa um estrutura auxiliar heap que armazena tuplas de nós visitados em um grafo, um vetor para armazenar os vértices 'pais' que são os vértices de menor custo conectados a um vértice visitado, um vetor para determinar os vértices inseridos e outro para determinar os "pesos" dos vértices. Enquanto o algoritmo de Kruskal utiliza apenas e estrutura de conjunto de vértices e de armazenamento de arestas ordenadas pelo peso.

Veja a diferença a partir do tempo e de alocação de média da execução do primeiro grafo 10 vezes: