

Trabalho de Análise de Algoritmos: Comparação dos Algoritmos de Árvore Geradora Mínima - Kruskal e Prim

- Em duplas até dia 28/10 via classroom
- Apenas um integrante da dupla deve postar os arquivos

Nesse trabalho cada dupla deve realizar uma análise comparativa dos algoritmos de árvore geradora mínima de Kruskal e Prim em termos de desempenho, avaliando seus tempos de execução em cenários diferentes. A análise deve ser feita com a implementação dos algoritmos utilizando as **linguagens C ou Python**.

Além disso, a análise deve considerar o comportamento dos algoritmos sobre grafos de diferentes tamanhos e densidades, considerando apenas a sua representação na forma de **lista de adjacência**.

Algoritmos a serem implementados

Os seguintes algoritmos de ordenação devem ser implementados:

- **Algoritmo de Kruskal**
 - Utilize a estrutura de dados [Union-Find](#) para as operações envolvendo conjuntos.
- **Algoritmo de Prim**
 - Utilize a estrutura de dados heap (binário) mínimo para implementar a fila de prioridades utilizada pelo algoritmo.

Você deve considerar a execução dos algoritmos em grafos **densos** (quando a quantidade de arestas é próxima do número máximo possível de arestas) e **esparsos** (quando a quantidade de arestas é próxima do número de vértices). Todos os grafos gerados devem ser **conexos**.

Dessa forma, a dupla deve elaborar uma ou mais funções que sejam capazes de **gerar grafos** com essas configurações e diferentes tamanhos (em termos de quantidade de vértices).

Requisitos do Trabalho

1. **Implementação dos Algoritmos:**
 - Todos os algoritmos mencionados acima devem ser implementados na linguagem C ou Python.
 - Para cada variação de execução, registre o tempo de execução usando funções apropriadas para captura de tempo. Considere grafos ponderados densos e esparsos com **10, 100, 500, 1000, 2000, 5000 e 10000 vértices**.

2. Comparação de Tempo:

- Construa **gráficos** com os tempos médios de execução para os algoritmos de Kruskal e Prim em função do número de vértices para cada tipo de grafo (esparso e denso).
- Analise o desempenho dos algoritmos em grafos com diferentes densidades e discuta quais fatores influenciam o tempo de execução de cada algoritmo.
- Compare os tempos obtidos com a complexidade teórica de cada algoritmo considerando que tanto **Kruskal e Prim possuem complexidade $O(E \lg V)$** . Discuta eventuais diferenças observadas entre os tempos práticos e as complexidades teóricas.

Entrega

1. Código-Fonte:

- Um **único arquivo** contendo a função principal e as demais funções dos algoritmos implementados.
- Todo o código deve estar **documentado com comentários** relevantes para sua compreensão, com explicações claras sobre o funcionamento de cada função.

2. Relatório:

- O relatório deve conter os gráficos de tempo de execução gerados, a análise dos resultados e uma explicação sobre o comportamento de cada algoritmo.
- Inclua uma **descrição teórica das estruturas Union-Find e Fila de Prioridades** utilizadas na implementação dos algoritmos principais.
- Aponte observações sobre a complexidade de tempo de cada algoritmo, baseando-se na teoria e nos resultados obtidos.
- Apresente uma **conclusão geral** sobre a eficiência relativa dos algoritmos de Kruskal e Prim para os tipos de grafos avaliados.