



## **CIÊNCIA DA COMPUTAÇÃO**

### **Algoritmos em Grafos**

#### **Relatório do Trabalho: Geração de Árvores Aleatórias**

**Professor: Marco Aurélio Lopes Barbosa**

**Aluno: Gabriel de Souza Vendrame**

**RA: 112681**



## **1. Introdução**

O trabalho sobre geração de árvores aleatórias tem como objetivo praticar a implementação na prática de algoritmos em grafos. Este trabalho aborda três algoritmos geradores de árvores aleatórias, o algoritmo baseado em passeio aleatório, o algoritmo de Kruskal e o algoritmo de Prim.

## **2. Desenvolvimento**

### **2.1. Linguagem de programação utilizada**

A linguagem de programação que foi utilizada durante o desenvolvimento do trabalho foi a linguagem Python na versão 3.9, pois é uma linguagem simples e de fácil compreensão.

### **2.2. Representação dos grafos e atributos**

Os grafos e vértices foram representados como classes de objetos e usando lista e matriz de adjacências.

### **2.3. Informações importantes**

Para o algoritmo de passeio aleatório foram utilizadas classes de objetos distintos em relação às usadas para o algoritmo de Kruskal e algoritmo de Prim, para facilitar na construção destes e evitar alterações desnecessárias nos códigos feitos anteriormente.

## **3. Resultados**

### **3.1. Configurações da máquina**

Ao longo do trabalho, os algoritmos foram construídos e executados em um computador desktop e em um notebook.

O desktop tem a seguinte configuração:

Processador Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz 3.90 GHz.

Memória RAM de 8,00 GB.

GPU NVIDIA GeForce GT 1030.

Sistema operacional de 64 bits, processador baseado em x64.



O notebook tem a seguinte configuração:

Processador Intel(R) Core(TM) i5-10210U CPU @ 1.60Hz 2.11GHz.

Memória RAM de 8,00 GB.

GPU NVIDIA GeForce MX 250.

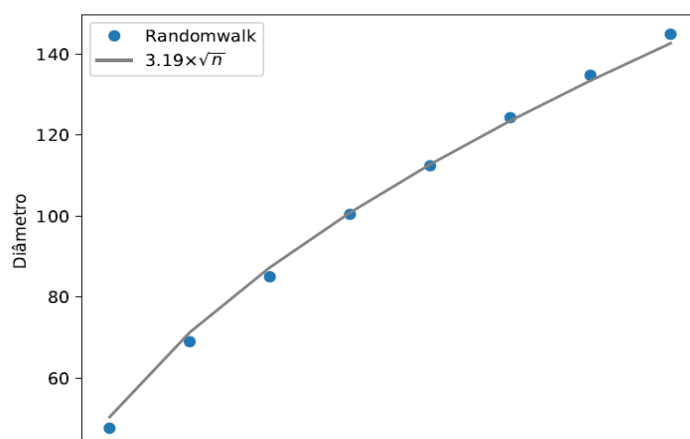
Sistema operacional de 64 bits, processador baseado em x64.

### 3.2. Tabela dos tempos de execução

n	Random Tree Walk	Random Tree Kruskal	Random Tree Prim
250	0,86 segundos	13,68 segundos	9,84 segundos
500	1,98 segundos	80,58 segundos	48,07 segundos
750	5,53 segundos	209,62 segundos	130,02 segundos
1000	6,44 segundos	419,06 segundos	269,58 segundos
1250	8,63 segundos	683,28 segundos	494,06 segundos
1500	9,28 segundos	1034,73 segundos	685,41 segundos
1750	10,38 segundos	1454,02 segundos	976,56 segundos
2000	14,23 segundos	1922,91 segundos	1178 segundos
Tempo Total	57,33 segundos	5817,90 segundos	3791,53 segundos

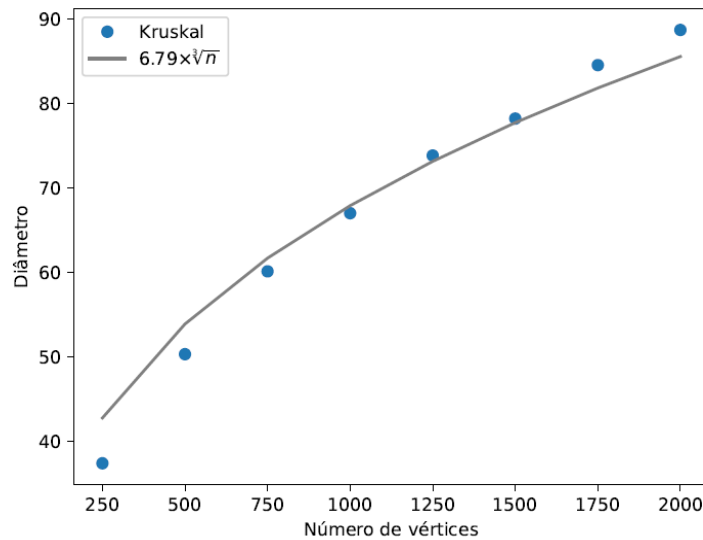
### 3.3. Gráficos do diâmetro médio

#### Random-tree-walk

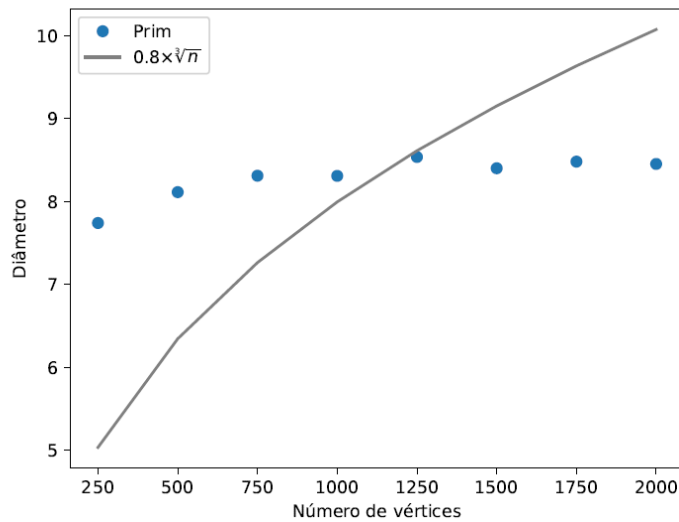




### Random-tree-kruskal



### Random-tree-prim



Os gráficos do algoritmo de passeio aleatório e do algoritmo de Kruskal mostram que os resultados obtidos estão dentro do esperado pela curva do gráfico, já o do algoritmo de Prim mostra um resultado totalmente fora do esperado.



#### **4. Experiência de desenvolvimento**

##### **4.1. Dificuldades encontradas e como foram superadas**

As maiores dificuldades que foram encontradas na execução do trabalho, foi a dificuldade de compreender como o grafo seria representado nos algoritmos e também dificuldade com a linguagem de programação utilizada, uma vez que eu não a utilizo muito, mas ao rever algumas das aulas gravadas e com a ajuda de colegas ambas dificuldades foram superadas.

##### **4.2. Destaques**

Foi importante descobrir que algumas funções simples, por exemplo a função pop, que remove primeiro elemento de uma lista, pode mudar o tempo de execução.

##### **4.3. Aprendizado final**

O trabalho sobre geração de árvores aleatórias mostra no final que, até mesmo problemas que possam parecer abstratos no começo, podem ser descritos em linha de código, além de reforçar a importância do auxílio das máquinas na resolução desse tipo de problema.