



Centro Universitário FEI

Projeto de CA-2330

Relatório - Parte 4

Grupo Nº 11

João Pedro Rosa Cezarino - R.A: 22.120.021-5

Lucca Bonsi Guarreschi - R.A: 22.120.016-5

Vitor Martins Oliveira - R.A: 22.120.067-8

**São Bernardo do Campo
2020**

1 Árvore

Em teoria dos grafos, uma **árvore** é um grafo conexo (um grafo é conexo se existir um caminho entre qualquer par de vértices) e acíclico (que não possui ciclos). Toda árvore é um grafo, mas nem todo grafo é uma árvore. Também é importante ressaltar que toda árvore é um grafo bipartido e planar.

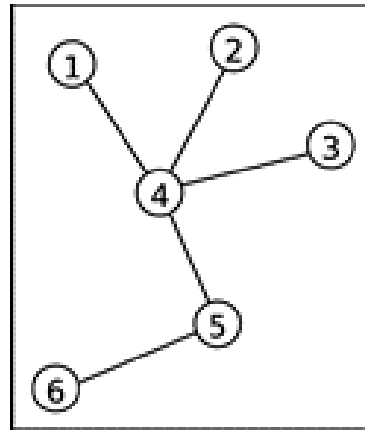


Figura 1

A união de árvores dá origem à uma **floresta**. Além disso, um grafo acíclico e desconexo também é chamado de floresta.

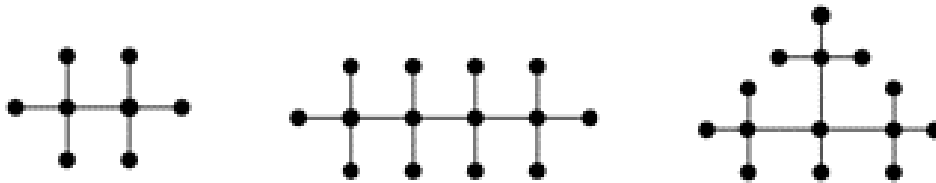


Figura 2

Se o grafo for considerado árvore, ele possuirá $n - 1$ arestas. O que fica evidente na imagem abaixo:

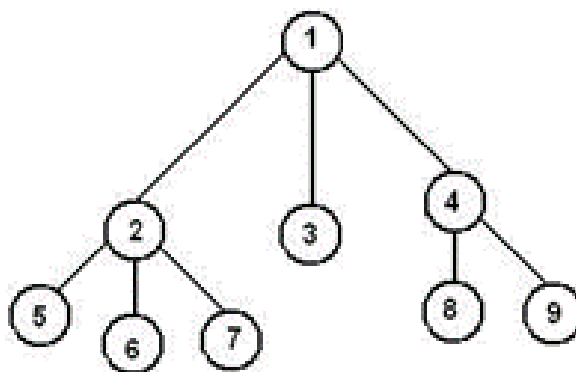


Figura 3

Uma **árvore geradora** é um subgrafo de um grafo G que contém todos os vértices de G . Como árvores são conexas, todo grafo dotado de árvore geradora é conexo. Reciprocamente, todo grafo conexo tem (pelo menos) uma árvore geradora.

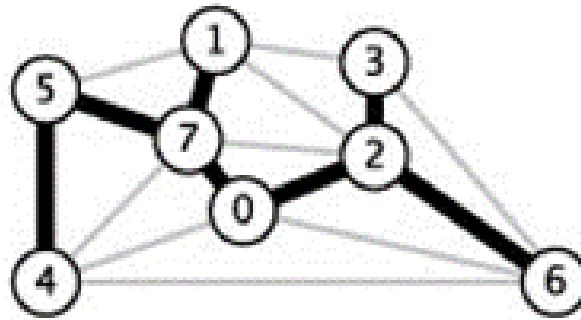


Figura 4

Uma **árvore geradora mínima** de G é qualquer árvore geradora de G que tenha custo mínimo. Ou seja, uma árvore geradora T de G é mínima se nenhuma outra árvore geradora tem custo menor que o de T . Existem alguns algoritmos já conhecidos para encontrar a árvore geradora mínima em um grafo, como por exemplo o **Algoritmo de Kruskal** e o **Algoritmo de Prim**.

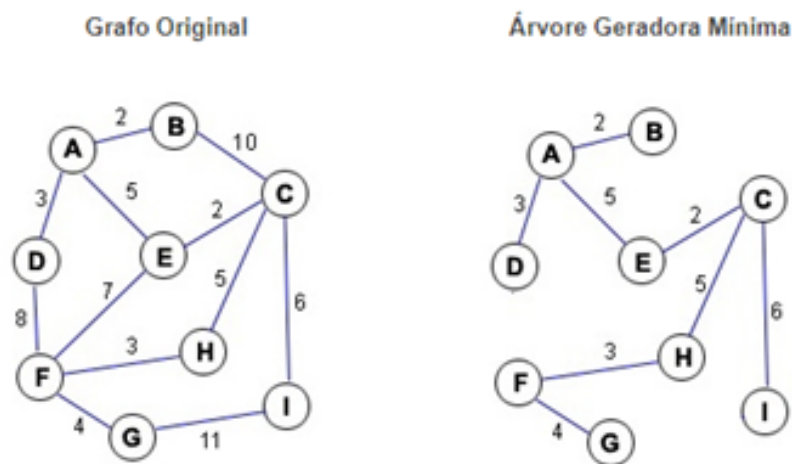


Figura 5

2 Descrição do Programa

Inicialmente o arquivo onde se encontra a matriz de adjacência "A.txt" é aberto e a leitura de cada linha da matriz é realizada, com o objetivo de retirar os espaços (" \n") e transformar o conteúdo em números inteiros (do tipo "int"). Após esse processo, o programa verifica e retira (se houver a presença) os espaços em branco desnecessários na matriz.

```

import networkx as nx
import numpy as np
import matplotlib.pyplot as plt

arq = open('teste2.txt', 'r') #abre o arquivo no modo read
lista = arq.readlines() #le o arquivo txt e cria uma lista do seu conteúdo
lista_final = []
arq.close()

for x in lista: #esse for le a lista o arquivo e adiciona o seus elementos á outra lista removendo os quebra linhas
    x = x.rstrip('\n')
    lista_final.append(x)

for num in lista_final: #remove os espaços em brancos do final da lista caso tenha algum
    if ' ' in lista_final:
        lista_final.remove(' ')
Matriz=[[int(num) for num in line.split(' ')] for line in lista_final] #transforma a lista em uma matriz

```

Figura 6

Então, Transforma-se a lista extraída do arquivo que contém a matriz de adjacência em um array que pode ser interpretado pela biblioteca *Numpy* e dessa forma podemos incorporar esta matriz aos módulos da biblioteca *NetworkX*.

Logo após a execução anterior, utiliza-se a função *is_tree* da biblioteca *NetworkX*, que retorna *TRUE* caso a quantidade de vértices seja diferente da quantidade de arestas mais um e *FALSE* se a condição anterior for falsa. Portanto, se a função retornar *TRUE* o grafo em questão é uma árvore, caso contrário, o grafo não é uma árvore.

```

A = np.matrix(Matriz)
G = nx.from_numpy_matrix(A)

nx.draw(G, with_labels=True, font_weight='bold')
plt.show()

print(f"\n***** RESULTADO *****\n")
def is_tree(G):
    if nx.number_of_nodes(G) != nx.number_of_edges(G) + 1:
        return print("O grafo não é uma árvore")
    else:
        return print("O grafo é uma árvore")
is_tree(G)

print("\n----- INTEGRANTES -----")
print("-----")
print("Nome: Lucca Bonsi Guarreschi / R.A: 22.120.016-5")
print("-----")
print("Nome: Vitor Martins Oliveira / R.A: 22.120.067-8")
print("-----")
print("Nome: João Pedro Rosa Cezarino / R.A: 22.120.021-5")
print("-----")
print("Turma: 010")
print("-----")

```

Figura 7

Por fim, obtém-se os resultados impressos no terminal e uma representação gráfica do grafo que foi submetido ao programa.

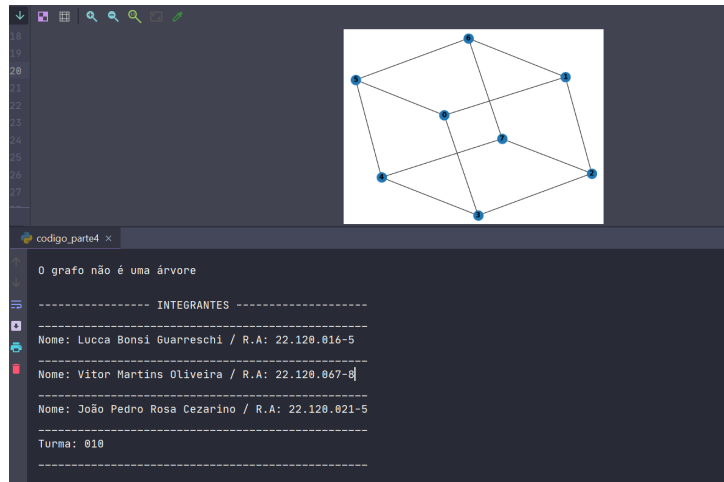


Figura 8

Atenção: Para o correto funcionamento do programa as bibliotecas *Matplotlib*, *NetworkX* e *Numpy* devem estar instaladas no sistema. Para instalá-las, execute os comandos abaixo no terminal da sua IDE ou na linha de comando do sistema:

```
pip install Matplotlib
pip install NetworkX
pip install Numpy
```

Também é importante lembrar que para instalar as bibliotecas acima, o instalador *pip* deve estar instalado no sistema em que o programa será executado. Para instalá-lo, deve-se seguir o passo a passo da documentação que está presente no link abaixo:

<https://pip.pypa.io/en/stable/installing/>

A biblioteca ***NetworkX*** é um pacote Python utilizado na criação, manipulação e estudo das estruturas, da dinâmica e das funções de redes complexas e de grafos.

A biblioteca ***Matplotlib*** tem a função de criar representações gráficas estáticas, animadas e interativas em Python.

NumPy é uma biblioteca para Python, que adiciona suporte para matrizes e arrays multidimensionais grandes, junto com uma grande coleção de funções matemáticas de alto nível para operar estes arrays.