



Centro Universitário FEI

Projeto de CA-2330

Relatório - Parte 5

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 Grafos e Trilhas de Euler

Um grafo G é dito ser euleriano se há uma trilha em G que contenha todas as suas arestas. Esta trilha é dita ser uma trilha euleriana. O grafo da figura abaixo por exemplo, é euleriano já que ele contém a trilha: $(u_1, u_2, u_3, u_4, u_5, u_3, u_1, u_6, u_2, u_7, u_3, u_6, u_7, u_1)$, que é euleriana.

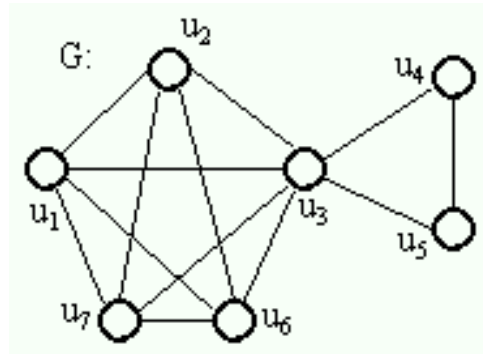


Figura 1

Um Grafo conexo G é Euleriano se cada vértice de G possui grau par, ou seja, se o grafo é euleriano todos os vértices têm grau par, e além disto, se todos os vértices do grafo têm grau par então o grafo é euleriano.

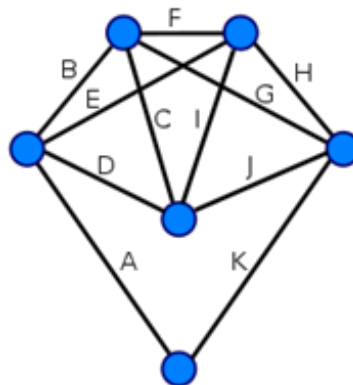


Figura 2

Outra condição que provê uma solução simples para determinar se um grafo é euleriano é a de que um grafo M é euleriano se, e somente se, M é conexo.

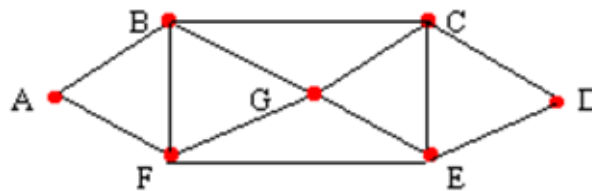


Figura 3

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.

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*.

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

arq = open('A.txt', 'r') #abre o arquivo no modo read
lista = arq.readlines() #le o arquivo txt e cria uma lista do seu conteudo
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
A = np.matrix(Matriz)
G = nx.from_numpy_matrix(A)
```

Figura 4

Logo após a execução anterior, utiliza-se a função *is_eulerian* da biblioteca *NetworkX*, que retorna *TRUE* caso o grafo for Euleriano e *FALSE* se não for. Portanto, se a função retornar *TRUE* o grafo em questão é Euleriano, caso contrário, o grafo não é. A função *has_eulerian_path* da biblioteca *NetworkX* é responsável por retornar *TRUE* caso o grafo possua um trilha de Euler e *FALSE* se não possuir. Se o grafo possuir uma trilha Euleriana ela será impressa no terminal junto com as outras informações.

```
if nx.is_eulerian(G) == True:
    print("O grafo é Euleriano.")
    x = list(nx.eulerian_circuit(G))
    print(x)
    nx.draw(G, with_labels=True, font_weight='bold')
    plt.show()
elif nx.is_eulerian(G) == False:
    print("O grafo não é Euleriano.")
    if nx.has_eulerian_path(G) == True:
        print(list(nx.eulerian_path(G)))
        nx.draw(G, with_labels=True, font_weight='bold')
        plt.show()
    else:
        print("O grafo não possui trilhas Eulerianas.")
        nx.draw(G, with_labels=True, font_weight='bold')
        plt.show()

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 5

Por fim, obtém-se os resultados impressos no terminal e uma representação gráfica do grafo que foi submetido ao programa. É importante ressaltar que as linhas impressas na cor vermelha não se tratam de um erro e sim um aviso de um dos módulos utilizados no programa, quanto à imagem do grafo gerada. Tal aviso não interfere na imagem gerada e nos resultados finais do código.

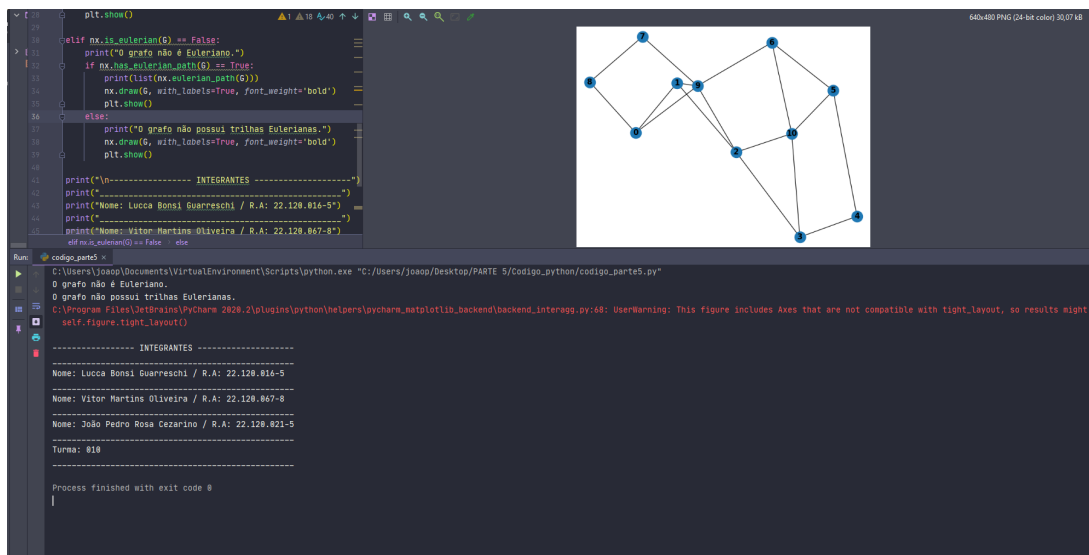


Figura 6

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.