

Aluna: Elisa Faria Almeida  
Matrícula: 2021006780  
Disciplina: Algoritmos e Grafos - SIN110

## Atividade 1 - 29 de agosto de 2022

### 1) Detalhes da implementação

A atividade em questão consiste na implementação de um programa que faça a leitura do arquivo de uma instância, imprima o nome da instância e suas dimensões e, posteriormente, salve o resultado em um arquivo de texto. A função *main*, responsável por executar o programa, foi estruturada da seguinte forma:

```
if __name__ == "__main__":  
    arquivo = sys.argv[1]  
    matriz = leituraArquivo(arquivo)  
    result = arquivo + ' ' + str(matriz.shape)  
    salvaResultado(result)  
    print(result)
```

Primeiramente, foi definida a função “leituraArquivo”, recebendo como parâmetro o nome da instância através de ‘sys.argv’, que referencia o nome do arquivo passado como parâmetro de compilação. Nessa função, o conteúdo do arquivo é lido e armazenado em uma matriz do tipo Numpy. Para isso, foi utilizada a função ‘loadtxt’ para retornar o array(matriz) Numpy a partir dos dados do arquivo de texto. A função “leituraArquivo” tem como retorno a própria matriz, conforme figura abaixo.

```
#função para ler o arquivo do tipo txt e converter a matriz para o tipo numpy  
def leituraArquivo(nome):  
    caminho = "C:/Users/faria/OneDrive/Área de Trabalho/ProjetosGrafos/SIN110_-_ATV1_-_Datasets_grafos/" + nome + ".txt"  
    with open(caminho, 'r') as arquivo:  
        matriz = np.loadtxt(arquivo)  
    return matriz
```

Após gerar a matriz do tipo numpy, a variável ‘result’ armazena o nome do arquivo e suas dimensões, que são calculadas por meio da função ‘shape’. Essa variável é passada como parâmetro para a função “salvaResultado” que cria um novo arquivo de texto e salva essas informações nele. Por fim, o nome e as dimensões da matriz (contidas em ‘result’) são impressas na tela.

```
#função para salvar o nome e dimensão da matriz em outro, informações que serão passadas como parâmetro  
def salvaResultado(result):  
    arq = open('C:/Users/faria/OneDrive/Área de Trabalho/ProjetosGrafos/Resultados/resultado.txt', 'a+')  
    arq.writelines(result + '\n')  
    arq.close()
```

## 2) Resultados

a) Dataset 1 - Nome: “exemplo”; Dimensão: 7x7.

```
ativ1 x
C:\Users\far...AppData\Local\Micro...
exemplo (7, 7)

Process finished with exit code 0
```

```
ativ1.py x resultado.txt x
1 exemplo (7, 7)
2
```

b) Dataset 2 - Nome: “ponte”; Dimensão: 4x4.

```
ativ1 x
C:\Users\far...AppData\Local\Micro...
ponte (4, 4)

Process finished with exit code 0
```

```
ativ1.py x resultado.txt x
1 exemplo (7, 7)
2 ponte (4, 4)
3
```

c) Dataset 3 - Nome: “zachary”; Dimensão: 34x34.

```
ativ1 x
C:\Users\far...AppData\Local\Micro...
zachary (34, 34)

Process finished with exit code 0
```

```
ativ1.py x resultado.txt x
1 exemplo (7, 7)
2 ponte (4, 4)
3 zachary (34, 34)
4
```

## 3) Dificuldades encontradas

A maior dificuldade para mim foi entender o processo de passar argumentos como parâmetro a partir da linha de comando e a utilização da função sys.argv.

Link do github: <https://github.com/elisafaria/Algoritmos-e-grafos/tree/main/Ativ01>