

CMAC03 – Algoritmos e Grafos

Relatório Atividade 1

Nome: Pablo Augusto matos da Silva

Matrícula: 2022015139

https://github.com/pabloaugmatrix/grafos/tree/main/AT1/preparacao_ambiente

Recursos externos:

Para a atividade proposta foram utilizadas duas bibliotecas externas:

- **numpy** para funcionalidades relacionadas a matriz:

```
matriz = np.genfromtxt(arquivo)
matriz.shape
```

- **sys** para entrada de parâmetro via comando de execução do programa pelo terminal:

```
main(str(sys.argv[1]))
python3 main.py instancia
```

Estrutura do programa:

O programa consiste de 4 módulos:

- **main**: core do programa, todo fluxo principal se encontra aqui.

```
1  import sys
2  from criarMatriz import criar_matriz
3  from obterDimensao import obter_dimensao
4  from resultado import resultado
5
6  def main(instancia):
7      matriz = criar_matriz(instancia)
8      dimensao = obter_dimensao(matriz)
9      resultado(instancia, dimensao)
10
11  if __name__ == '__main__':
12      main(str(sys.argv[1]))
13
```

- **criarMatriz**: contém a função **criar_matriz** que recebe como parâmetro o nome de uma instância abre o arquivo referenciado e é realizada a leitura dos dados para gerar a matriz com a biblioteca numpy.

```
1  import numpy as np
2
3  def criar_matriz(instancia):
4      caminho = '/mnt/d/faculdade/2024_1/grafos/at1/'+ instancia + '.txt'
5      with open(caminho, 'rb') as arquivo:
6          matriz = np.genfromtxt(arquivo)
7      return matriz
8
```

- **obterDimensao:** contém a função **obter_dimensao** que recebe uma matriz como parâmetro e retorna uma tupla com suas dimensões obtidas através do método **shape** da biblioteca numpy.

```
1 import numpy as np
2
3 def obter_dimensao(matriz):
4     return matriz.shape
5
```

- **resultado:** contém a função **resultado** que recebe como parâmetro o nome da instância, a matriz gerada e suas respectivas dimensões, para então imprimir os dados na tela e por fim salvá-los em um arquivo.

```
1 def resultado(instancia , dimensao):
2     print("Instancia:", instancia)
3     print("Linhas:", dimensao[0])
4     print("Colunas:", dimensao[1])
5     with open('resultado.txt', 'w') as arquivo:
6         arquivo.write(f'Instancia: {instancia}\n')
7         arquivo.write(f'Linhas: {dimensao[0]}\n')
8         arquivo.write(f'Colunas: {dimensao[1]}\n')
9
```

Considerações finais:

Para realização da atividade foram consultadas as seguintes fontes:

- numpy.org: documentação a respeito dos metodos contidos na biblioteca numpy;
- chat.openai.com: pesquisa referente semantica da linguagem python;
- stackoverflow.com: pesquisa a respeito das convenções para nomeação de funções em python.

Para testar o programa foi utilizado o arquivo **ponte.txt** fornecido pelo professor.

```
at1 > ≡ ponte.txt
1  0 2 2 1
2  2 0 0 1
3  2 0 0 1
4  1 1 1 0
5  |
```

```
pabloaugmat@Thoth:/mnt/d/faculdade/2024_1/grafos/at1$ python3 main.py ponte
Instancia: ponte
Linhas: 4
Colunas: 4
```

```
at1 > ≡ resultado.txt
1  Instancia: ponte
2  Linhas: 4
3  Colunas: 4
4  |
```