SIN110 – Algoritmos e Grafos
Matheus Luz - 2020032426
ATIVIDADE 1 (ATV1) : Entrega 29/08
Código: Github

A proposta é criar um protótipo de software que faça a leitura de um arquivo de uma dada instância, exiba os resultados obtidos e salve estes em um outro arquivo. No caso, foi necessário o uso das funções da biblioteca Numpy para a criação de matrizes e para a obtenção das dimensões destas.

O código foi estruturado conforme as indicações de organização do Slide 1, ou seja, dividindo as etapas / funções do software em pastas e arquivos para agilizar o desenvolvimento e facilitar o processo de modularização.

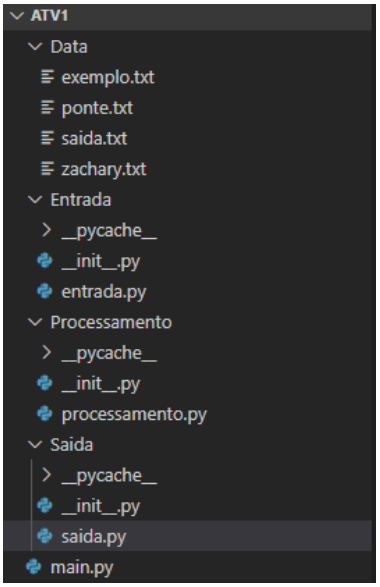

Figura 1. Estrutura do código.

Como pode ser observado, o código foi dividido em 4 pastas (além do arquivo main.py), sendo estas Data, Entrada, Processamento e Saída.

Pasta Data

É onde armazeno os arquivos que serão lidos pelo software e a saída (saida.txt) que armazena os resultados obtidos após o processamento das matrizes e obtenção das suas dimensões.

Pasta Entrada

É onde armazeno o arquivo responsável por ler o conteúdo das instâncias e retornar uma matriz numpy como resposta para a main. O nome do arquivo que será aberto é

recebido como parâmetro e será usado após a concatenação para encontrar este no disco. O arquivo é aberto, a função “loadtxt” do Numpy é usada e o resultado é armazenado na variável matriz. Por fim, o arquivo é fechado e a matriz é retornada.

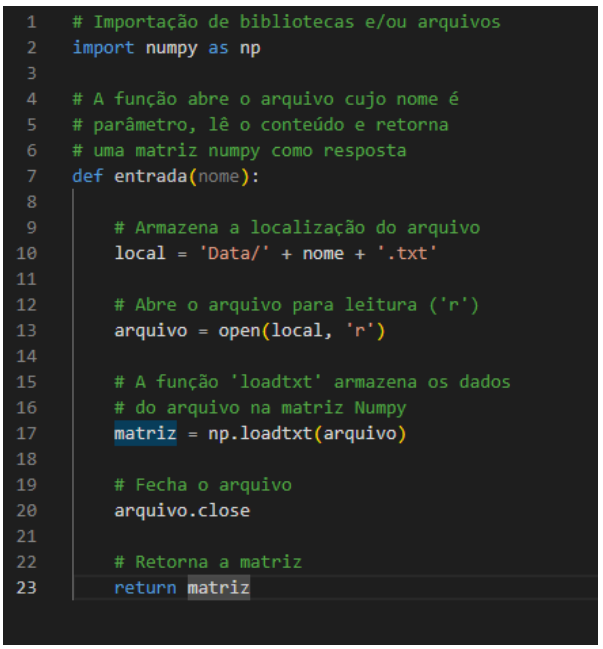

Figura 2. Arquivo entrada.py

Pasta Processamento

É onde armazeno o arquivo responsável por processar a matriz que foi obtida anteriormente no arquivo de entrada. O nome do arquivo que foi aberto e a matriz obtida são usados de parâmetro, então a quantidade de linhas e colunas são obtidas através da função “len” e são armazenadas nas variáveis de nome iguais. Por fim, o código exibe o nome do arquivo e as dimensões da matriz e retorna as linhas e colunas para a main.

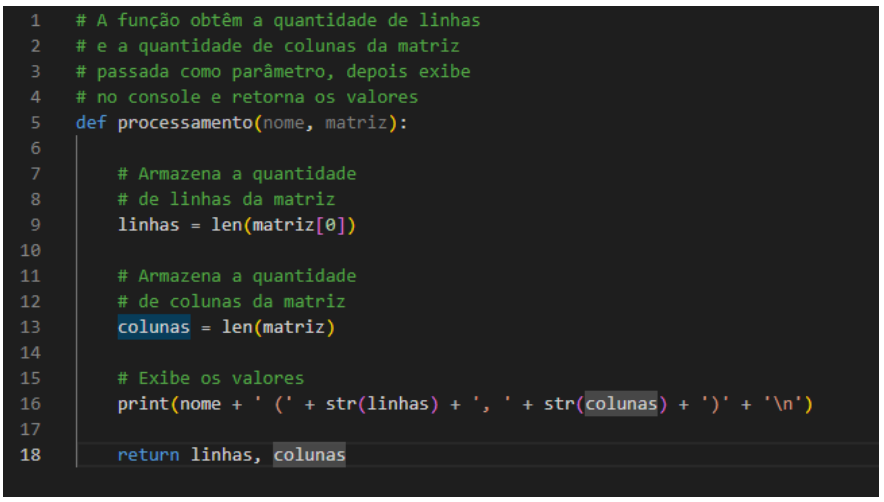

Figura 3. Arquivo processamento.py

Pasta Saída

É onde armazeno o arquivo responsável por armazenar os resultados obtidos através do processamento da matriz Numpy. De modo simples, a função recebe o nome do arquivo que foi lido, a quantidade de linhas e colunas, então armazena estes no arquivo de saída e fecha este

```python
# A função é responsável por armazenar
# os resultados obtidos do processamento
# da matriz Numpy. Os resultados são
# as dimensões da matriz, ou seja,
# a quantidade de linhas e colunas
def saida(nome, linhas, colunas):

    # Abre o arquivo para escrita ('a+')
    arquivo = open('Data/saida.txt', 'a+')

    # Faz a escrita no arquivo
    arquivo.write(nome + ' (' + str(linhas) + ', ' + str(colunas) + ')' + '\n')

    # Fecha o arquivo
    arquivo.close
```

Figura 4. Arquivo saida.py

Arquivo main.py

É onde fica a função principal do software, nesta os arquivos necessários para o funcionamento são inicialmente importados. Dentro da função principal, foi criado um laço para processar os três arquivos “.txt” que deveriam ser usados como exemplo. Então as funções de entrada, processamento e saída são chamados em ordem respectiva.

```python
# Importação de bibliotecas e/ou arquivos
import Processamento as p
import Entrada as e
import Saida as s

# Função principal
if __name__ == '__main__':

    # Loop para automatizar os testes
    for i in range(3):

        # Condicional para definir
        # qual arquivo será aberto
        if i == 0:
            nome = 'exemplo'

        elif i == 1:
            nome = 'ponte'

        else:
            nome = 'zachary'

        # 'matriz' é a matriz que será
        # obtida a partir da função de entrada
        matriz = e.entrada(nome)

        # A função obtêm a quantidade de linhas
        # e a quantidade de colunas da matriz,
        # exibe as dimensões obtidas na tela e
        # retorna estes valores
        linhas, colunas = p.processamento(nome, matriz)

        # A função
        s.saida(nome, linhas, colunas)
```

Figura 5. Arquivo main.py