

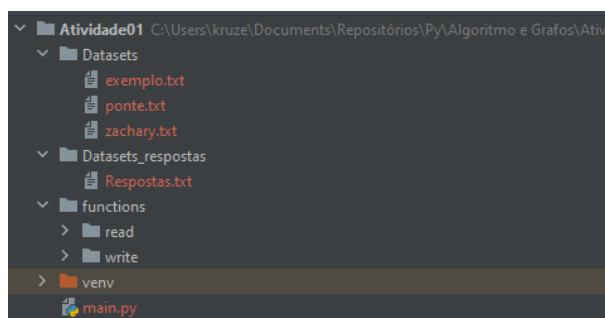
RELATÓRIO DE EVIDÊNCIAS DE IMPLEMENTAÇÃO ATV₁

Victor Kruze Fiori - 2021005488

24/08/2022

1 Desenvolvimento inicial

Primeiramente organizei os diretórios de arquivos de acordo com as suas funções, nessa atividade, em funções para escrita e leitura de dados assim como para armazenamento dos arquivos a serem lidos pelo sistema e o arquivo para armazenar as respostas



Diretórios

Após criado o esqueleto a ser utilizado, comecei modulando as funções que seriam necessárias para a resolução do problema, pelo arquivo "main.py", pensei nas funções leitura_matriz para leitura e salva_dado para escrita, e no começo não me preocupei em receber os arquivos por parâmetros e deixei pré-definido o arquivo "exemplo.txt" para ser lido e interpretado pelo sistema em todos os testes.

2 Desenvolvimento de Funções

Como dito anteriormente, foi preciso de duas funções, a primeira pra ler e interpretar a matriz recebida pelo dataset a segunda para escrever as informações desejadas.

Para sobrepor a leitura e interpretação do arquivo importei o módulo Numpy e utilizei duas funções principais para isso, sendo elas: loadtxt e shape.

A função loadtxt carrega o arquivo e suas informações como número de linhas e número de colunas e sua documentação foi encontrada em: <https://numpy.org/doc/stable/reference/generated/numpy.loadtxt.html>, já a função shape retorna o número de linhas e colunas do array enviado como parâmetro em forma de uma tupla e sua documentação foi encontrada em: <https://numpy.org/doc/stable/reference/generated/numpy.shape.html>.

```
def leitura_matriz(dataset):
    caminho_data_set = "Datasets/" + dataset + ".txt"
    matriz = np.loadtxt(caminho_data_set)
    info = np.shape(matriz)
    return info
```

Já a função de escrita, de uma forma mais simples não precisou da importação de nenhum módulo, ela apenas abre o arquivo onde deverá ser feito a escrita dos dados recebidos como parâmetro e escreve esses dados e por fim fecha o arquivo finalizando a execução

```
def salva_dado(resultado):
    file = open('Datasets_respostas\\Respostas.txt', 'a+')
    file.writelines(resultado+'\n')
    file.close()
```

3 Finalização

Após as funções devidamente criadas e funcionando, comecei a entender e desenvolver o envio do nome do arquivo como parâmetro, consegui através do módulo sys apenas utilizando a sintaxe "sys.argv[1]", os argumentos recebidos na main sempre são a partir da 2ª posição do array (posição 1) já que a primeira é reservada para o nome do arquivo main.

Após o módulo implementado, resolvi colocar um passo de segurança para a leitura do argumento, colocando o arquivo "exemplo.txt" como base caso não fosse enviado nenhum parâmetro, por esquecimento ou qualquer outro motivo, e caso fosse enviado o sistema prioriza a leitura do recebido como parâmetro.

```
if __name__ == '__main__':
    path = 'exemplo'
    if len(sys.argv) > 1:
        path = sys.argv[1]
    res = leitura_matriz(path)
    salva_dado(str(path) + ' ' + str(res[0]) + ' ' + str(res[1]))
```

4 Dificuldades:

Tive três complicações principais, a primeira foi que não estava entendendo o funcionamento de enviar parâmetros pela main e não estava sabendo realizar a execução disso na IDE. Já a segunda dificuldade, tive na organização de diretórios e referenciamento deles pelo python, python apesar de ser uma linguagem user-friendly e de fácil aprendizado, não a utilizo com bastante frequência e acabo me perdendo em algumas sintaxes. Já a última foi com a instalação do módulo Numpy que tive problemas com o pip install e não estava entendendo o funcionamento correto, mas isso foi sobreposto com pesquisas também.

Fora essas, a leitura de documentação da biblioteca Numpy foi de fácil compreensão e a IDE utilizada foi bastante amigável para o desenvolvimento.

5 GitHub

Atividade disponível no github em <https://github.com/Kruzeffiori/Python/tree/main/Algoritmo%20e%20Grafos/Atividade%2001>