

Manipulação de Grafos

Trabalho Prático 1

Artur Papa - 3886¹, Vinícius Mendes- 3881², Jhonata Miranda-3859³

¹Universidade Federal de Viçosa - Campus Florestal

artur.papa@ufv.br, vinicius.o.mendes@ufv.br, jhonata.miranda@ufv.br

1. Introdução

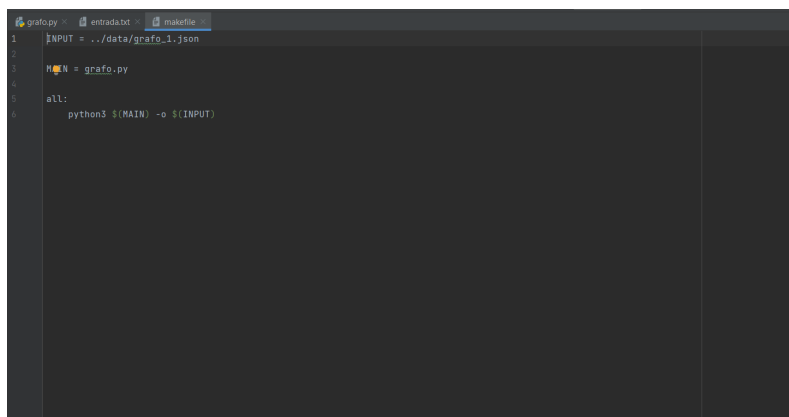
O trabalho em questão tem como objetivo abordar os conteúdos trabalhados na disciplina de Teoria e Modelo de Grafos, em específico, como visto nas aulas teóricas, a determinação de ordem, tamanho, densidade entre outros detalhes que podem ser observado a partir de grafos. A priori, é válido dizer que há a necessidade de identificar as características principais de cada grafo, dentre elas, se ele é ponderado, não direcionado e se possui ciclos, para que assim, as ações necessárias sejam tomadas para manipulação dele.

Neste momento, percebemos a necessidade de aplicar as implementações repassadas durante as aulas, para que assim, pudéssemos realizar buscas em largura e em profundidade, bem como, verificar se o mesmo é euleriano.

2. Desenvolvimento

2.1. Compilação e execução

Para a compilação dos arquivos-fonte, deve se usar o comando **make** para executar o makefile que está dentro do caminho **"TP01Grafo/src"**. Após isso, será exibido um menu, levando em consideração o grafo contido no arquivo de entrada definido na linha INPUT do makefile assim como mostra a figura 2, onde teremos opções para executar as funções e obter informações sobre o grafo. Para utilizar um arquivo .json obtido no site <https://paad-grafos.herokuapp.com/> deve-se alterar nome do arquivo de entrada no makefile.



```
1 INPUT = ../data/grafos_1.json
2
3 MAIN = grafo.py
4
5 all:
6 python3 $(MAIN) -o $(INPUT)
```

Figure 1. Makefile

Ao executar as funções, os resultados serão impressos no terminal, além da opção 11 que escreve seus resultados no arquivo **"TP01Grafo/out/saida.txt"**.

```
TRABALHO GRAFOS
-----
[0] SAIR

[1] Retornar a ordem do grafo
[2] Retornar o tamanho do grafo
[3] Retornar a densidade do grafo
[4] Retornar os vizinhos de um vertice fornecido
[5] Determinar o grau de um vértice fornecido
[6] Verificar se um vértice é articulação
[7] Determinar a sequência de vértices visitados na busca em largura e informar a(s) aresta(s) que não faz(em) parte da árvore de busca em largura
[8] Determinar o número de componentes conexas do grafo e os vértices de cada componente
[9] Verificar se um grafo possui ciclo
[10] Determinar a árvore geradora mínima de um grafo
[11] Verificar se um grafo é euleriano

Escolha ->
```

Figure 2. Exemplo de execução

2.2. Decisões Importantes

É válido salientar que optamos por utilizar a linguagem Python por sua grande gama de funções que poderiam nos auxiliar, a quantidade de fóruns especializados nessa linguagem, e por sua facilidade de programação, visto que, todos nós já tínhamos a utilizado.

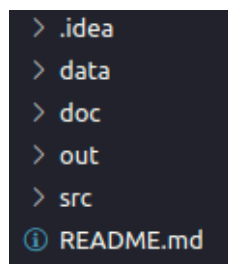


Figure 3. Exemplo de disposição das pastas

Diante disso, para melhor organização do projeto, definimos a necessidade da separação dos arquivos através de pastas. Isso por sua vez é responsável por uma melhor disposição do código, entendimento do projeto, bem como sua apresentação.

3. Conclusão

Posto isso, conclui-se que o trabalho em questão foi desenvolvido conforme o esperado, atingindo as especificações requeridas na descrição do mesmo em implementar os algoritmos em grafos postos em sala.

Por conseguinte podemos ressaltar que o material apresentado na disciplina foi de suma importância para o desenvolvimento do projeto, haja vista que as explicações dadas pelo professor nos ajudaram bastante a entender as etapas a serem executadas, a construção dos códigos e na disponibilização de alguns algoritmos de busca. Além disso podemos ressaltar que vídeos como os disponibilizados na disciplina de Teoria e Modelo de Grafos, foram de grande ajuda para o entendimento dos conceitos necessários para o desenvolvimento do projeto.

Em adição, é válido dizer que apesar das dificuldades na implementação do código, o trio foi capaz de superar e corrigir quaisquer erros no desenvolvimento do algoritmo. Por fim, verificou-se a assertiva para o objetivo do projeto em implementar os métodos de ordenação para inteiros e estruturas de dados.