

Relatório - REO 6

Arthur Henrique Sousa Cruz

Universidade Federal de Lavras

1 Execução

Para compilar e executar o código utilize os seguinte comando:

```
python3 main.py <arquivo-de-entrada> <arquivo-saida> <modo-de-representacao>
```

O modo de representação pode ser MA (para matriz adjacência) ou LA (para lista de adjacência).

2 Organização de arquivos

Os arquivos deste projeto foram organizados da seguinte forma:

- **gerador_aleatorio.py**: arquivo que implementa um gerador de grafos aleatórios;
- **main.py**: arquivo principal;
- **gerenciador_arquivo.py**: contém a implementação da leitura do arquivo de entrada;
- **grafo.py**: arquivo com as funções de um grafo;
- **lista_adj.py**: arquivo que implementa uma lista de adjacência;
- **matriz_adj.**: arquivo que implementa uma matriz de adjacência;
- **entradas**: pasta com arquivos de exemplo de entrada;
- **saidas**: pasta com arquivos de exemplo de saída (foram executados os arquivos *teste_1.txt*, *teste_2.txt*, *teste_3.txt* para obtenção das saídas).

3 Implementação

A implementação foi feita em *Python3*. O cálculo do grau médio foi feito com a seguinte fórmula:

$$\sigma = \frac{2 | E |}{| V |} ,$$

sendo σ o grau médio, e V e E os conjuntos de vértices e arestas, respectivamente.

4 Análise do algoritmo

Para comparar as estruturas implementadas foram realizados experimentos computacionais. As instâncias utilizadas encontram-se na pasta *entradas*. A instância *collaboration_graph_2.txt* foi criada apenas para auxiliar na implementação. O programa executou a instância *collaboration_graph.txt* por mais de 10 minutos e não obteve resultados. A Tabela 1 mostra as quantidade de vértices (V) e de arestas (E) das instâncias utilizadas para testes.

Tabela 1: Instâncias utilizadas para experimentos práticos.

	V	E
teste_1	5000	1000000
teste_2	2000	750000
teste_3	6000	1000000

As Tabelas 2 e 3 mostram os resultados obtidos para os experimentos utilizando matriz de adjacência e lista de adjacência respectivamente. A coluna σ apresenta o grau médio do grafo e a coluna ω o espaço em MB ocupado pelos grafos. Nota-se que a matriz de adjacência tem um custo muito maior de memória em relação à lista. Para a segunda instância, a matriz ocupa aproximadamente quase nove vezes mais espaço que a lista, enquanto para a primeira e terceira instância gasta 36 e 51 vezes mais. Isso a complexidade de espaço da matriz de adjacência: $O(|V|^2)$, assim como a da lista: $O(|V| + |E|)$. Devido a isso, exceto em grafos completos, o espaço ocupado por uma lista de adjacência tende a ser menor se comparado a uma matriz, e o experimento é um exemplo prático disso.

Tabela 2: Resultados para Matriz de Adjacência.

	σ	ω
teste_1	400	579.68
teste_2	750	96.73
teste_3	333.33	831.50

Tabela 3: Resultados para a Lista de Adjacência.

	σ	ω
teste_1	400	16.58
teste_2	750	12.17
teste_3	333.33	16.67