



**Universidade Federal de Santa Catarina**

Departamento de Informática e Estatística (INE)

Curso: Ciência da Computação

Disciplina: Grafos (INE5413)

Professor: Rafael de Santiago

**Estudante:** Guilherme Adenilson de Jesus (22100620)

## **Relatório A1 - Grafo não-dirigido**

A implementação do grafo não-dirigido e seus algoritmos foi feito na linguagem de programação Python, devido ao maior conhecimento sobre sua *syntax* e estruturas. A seguir, é apresentada uma pequena justificativa das decisões feitas para uma das questões exigidas.

### 1. Representação

A montagem do grafo foi feita a matriz de adjacência utilizando um “dicionário dentro de um dicionário”. Se assemelha a uma matriz de comum, mas possibilitando o uso de índices do tipo *string*, facilitando de certa forma a criação a partir do arquivo oferecido e na aplicação de alguns algoritmos. Contudo, essa decisão implicou numa maior complexidade na busca do grau e vizinhos de um vértice ( $O(n)$ ), pois será necessário passar pelos vértices uma vez. Ademais, foi armazenada uma lista de arestas para ser utilizada em alguns algoritmos, invés de ser necessário fazer uma busca na matriz de adjacência toda vez que precisasse.

### 2. Busca

Para a busca em largura, as variáveis Conhecida, Distância e Antecessor também utilizaram o tipo dicionário, principalmente para manter um padrão com o escolhido para representação. Ademais, a fila foi implementada numa classe Queue, e seu conteúdo armazenado numa lista. Ainda, foi introduzido mais uma estrutura chamada Níveis, que separa os vértices em camadas para ser utilizado no momento de imprimir o resultado no formato exigido no roteiro.

### 3. Ciclo Euleriano

Como padrão, a variável Conhecida é um dicionário. Também foi feito uma adaptação na análise das arestas, pois na representação só foi decidida armazenar conforme lido no arquivo, necessitando uma verificação para outro sentido.

### 4. Bellman-Ford

A decisão de utilizar o Bellman-Ford, e não o Dijkstra, foi pela maior facilidade de implementação do algoritmo, mas também por ser mais completo (embora tenha uma maior complexidade). Como nos exercícios anteriores, foi utilizado dicionário para variáveis Distância e Antecessor para manter o padrão com

a representação. Ainda, como no Ciclo Euleriano, foi feita uma adaptação para verificação do outro sentido da aresta.

## 5. Floyd-Warshall

Por padrão, as matrizes Distância e Antecessor (“dicionário dentro de dicionário”) foram inicializadas com valores infinitos e nulos, respectivamente. Após suas criações foram atribuídos os dados corretos como no algoritmo original. Essa decisão foi feita para evitar exceções de indexação incorreta pela forma que o Python cria as chaves e seus conteúdos na estrutura. O restante da implementação não necessitou de adaptação.