

UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC
CENTRO TECNOLÓGICO – CTC
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA – INE

RELATÓRIO T1

DISCIPLINA: GRAFOS - INE5413
PROFESSOR: RAFAEL DE SANTIAGO

ALUNOS: ARTHUR BARICHELO (16200636)
LETICIA DO NASCIMENTO (16104595)
MARCOS AURÉLIO LESSA (17100526)

FLORIANÓPOLIS, 19 DE SETEMBRO DE 2019

1. Introdução

Para a realização do trabalho, escolhemos Python como linguagem de programação para o desenvolvimento, por questão de afinidade dos membros da equipe com a mesma.

Optamos por criar o Grafo como sendo uma *namedtuple*, uma espécie de estrutura em Python. Como se pode observar na imagem abaixo, esta estrutura nomeada de *Graph*, tem como atributos: *num_vertices*, *conj_vertices*, *conj_arestas*, que, respectivamente, representam o numero de vértices presente no Grafo, o conjunto de vértices, e o conjunto de arestas do mesmo.

```
# Cria o grafo propriamente dito
Graph = namedtuple("Graph", "num_vertices conj_vertices conj_arestas")
```

2. Vértices

Um vértice foi definido como uma tupla, onde o primeiro elemento indica o índice do vértice e o segundo elemento o seu rótulo. Na imagem abaixo, o vértice indica que seu índice é 1 e seu rótulo 'Itajai'.

```
(1, 'Itajai')
```

Já o conjunto de vértices, é uma lista que tem como tamanho igual ao atributo *num_vertices*. Esta lista contém todos os vértices do Grafo lido, como por exemplo:

```
[(1, 'Itajai'), (2, 'Camboriu'), (3, 'Brusque'), (4, 'Nova Trento'), (5, 'São João Batista'), (6, 'Navegantes')]
```

3. Arestas

Decidimos criar uma lista bidimensional para o conjunto das arestas, onde cada lista interna contém os vértices vizinhos de um determinado vértice. Ou seja, dado o exemplo dos vértices acima, a primeira posição da lista externa conterá todos os vértices com os quais o vértice (1, 'Itajai') se conecta, formando uma aresta. Optamos fazer assim, pois, quando for necessário receber os vértices vizinhos de algum outro, acessamos a posição inicial da sua lista de vizinhos em $O(1)$, e resgatamos todos os vértices vizinhos a ele em $O(n)$, sendo n o número de vértices os quais se conectam com ele. Essa solução é vista com muito mais eficiência do que se tivéssemos criado uma lista com todas as arestas existentes no Grafo, pois para um Grafo com muitas arestas, levaria muito mais tempo para conseguir acessá-las.

```
[[(2, 'Camboriu'), 1.0], [(6, 'Navegantes'), 1.0]],
```

A imagem acima mostra a primeira posição da lista externa, representada pelo vértice (1, 'Itajai'). A lista interna nessa posição, indica os vértices com os quais o vértice (1, 'Itajai') se conecta, formando uma aresta. Cada elemento que se conecta com o vértice mencionado é representado por uma tupla, onde na primeira posição se encontra o vértice e na segunda o peso da aresta que conecta os dois vértices.