

Teoria dos Grafos - Algoritmo de Dijkstra

Integrantes do Grupo:

- Igor Benites Moura - 10403462
- Rodrigo Machado de Assis Oliveira de Lima - 10401873

Implementado em Python

Código fonte:

```
def dijkstra(self, origin):  
  
    # apenas para grafos ponderados  
  
    if not self.isWeighted:  
  
        print("Grafo não é ponderado")  
  
        return  
  
    # inicializa as variaveis  
  
    d = [float('inf')] * self.n  
  
    d[origin] = 0  
  
    A = list(range(self.n))  
  
    F = []  
  
    S = [origin]  
  
    # substituicao de 0 por -1 para facilitar a comparacao em forma  
de matriz  
  
    rot = [-1] * self.n  
  
    # enquanto A não for vazio  
  
    while A:
```

```
# pega o menor valor de d

r = A[0]

for i in A:

    if d[i] < d[r]:

        r = i

# atualiza os nos visitados

F.append(r)

A.remove(r)

# olha os nos adjacentes de r e atualiza os valores de d
para o menor caminho

S = [i for i in A if self.adj[r][i] != float('inf')]

for i in S:

    p = min(d[i], d[r] + self.adj[r][i])

    if p < d[i]:

        d[i] = p

        rot[i] = r

return d, rot
```

Resolução:

```
from TGrafoDijkstra import Grafo

g = Grafo(4, isWeighted=True)

g.inserA(0,1,20) # insere aresta 0->1 com peso 20
g.inserA(0,2,30)
g.inserA(0,3,50)
g.inserA(1,0,20)
g.inserA(1,2,40)
g.inserA(1,3,15)
g.inserA(2,0,30)
g.inserA(2,1,40)
g.inserA(2,3,15)
g.inserA(3,0,50)
g.inserA(3,1,15)
g.inserA(3,2,15)

d, rot = g.dijkstra(2)
print("Distâncias: ", d)
print("Rotas: ", rot)
```

```
Distâncias: [30, 30, 0, 15]
Rotas: [2, 3, -1, 2]
```

Resolvemos implementar o algoritmo em grafo orientado por ser mais versátil e também funcionar com grafos não orientados, considerando que o oposto não é verdadeiro.

Tendo em conta que substituímos cada número de cidade para um a menos para trabalhar com index de arrays e matrizes (e a rota para o próprio vértice para -1), podemos concluir que:

- A distância ótima da cidade 3 até a cidade 1 é:

3 -> 1, com distância de 30km

- A distância ótima da cidade 3 até a cidade 2 é

3 -> 4 -> 2, com distância de 30km

- A distância ótima da cidade 3 até a cidade 4 é

3 -> 4, com distância de 15km