

# Teoria dos Grafos - Algoritmo de Bellmann-Ford

## Integrantes do Grupo:

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

Implementado em Python

## Código fonte:

```
def bellmannFord(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  
  
    rot = [-1] * self.n  
  
  
    for j in range(self.n):  
  
        for i in range(self.n):  
  
            #se tiver um caminho de j para i  
  
            if self.adj[j][i] != float('inf'):  
  
                #se a distancia da origem ate i for maior que a  
                distancia de j + peso da aresta  
  
                if d[i] > d[j] + self.adj[j][i]:
```

```

        # atualiza a distancia de i

        d[i] = d[j] + self.adj[j][i]

        rot[i] = j

    return d, rot

```

### Resolução:

Lembrando 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)

#### 1) Grafo do material de aula:

```

# 1) Grafo do material de aula (slides)
g1 = Grafo(5, isWeighted=True)

g1.insereA(0,1,1) # insere aresta 0->1 com peso 6
g1.insereA(0,4,1)
g1.insereA(1,2,1)
g1.insereA(1,3,2)
g1.insereA(2,3,4)
g1.insereA(2,4,2)
g1.insereA(3,0,3)
g1.insereA(4,0,2)
g1.insereA(4,3,1)

d, rot = g1.bellmannFord(0)
print("Grafo 1: ")
print("Distâncias: ", d)
print("Rotas: ", rot)

```

```

Grafo 1:
Distâncias: [0, 1, 2, 2, 1]
Rotas: [-1, 0, 1, 4, 0]

```

(Rota equivalente à [0, 1, 2, 5, 1] da atividade nos slides)

2) Grafo da atividade anterior:

```
# 2) Grafo da atividade anterior
g2 = Grafo(4, isWeighted=True)

g2.insereA(0,1,20) # insere aresta 0->1 com peso 20
g2.insereA(0,2,30)
g2.insereA(0,3,50)
g2.insereA(1,0,20)
g2.insereA(1,2,40)
g2.insereA(1,3,15)
g2.insereA(2,0,30)
g2.insereA(2,1,40)
g2.insereA(2,3,15)
g2.insereA(3,0,50)
g2.insereA(3,1,15)
g2.insereA(3,2,15)

d, rot = g2.bellmannFord(2)
print("\nGrafo 2: ")
print("Distâncias: ", d)
print("Rotas: ", rot)
```

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