

# Relatório - Trabalho 2 - AEDs 3

Eduardo Sapio Saccardo  
2019.1.08.003

Caio Marinello Moura Leite  
2019.1.08.032

22/07/2022

## Descrição do problema e abordagem utilizada

O problema consiste em achar o diâmetro de um grafo dado, ou seja, achar o caminho mais curto entre os nós mais distantes do grafo. Para a resolução, utilizamos o algoritmo de Floyd Warshall para descobrir os dois vértices mais distantes do grafo e, em cima desses dois vértices, utilizamos o algoritmo de Dijkstra para encontrar todos os vértices que fazem parte desse caminho.

## Pseudo-códigos utilizados

### Floyd Warshall(quantVertices)

```
for i ate quantVertices
  for j ate quantVertices
    dist[i][j] = mat[i][j] // Mat é a saída do floyd warshall
                          // Assim, temos a resposta replicada em dist

for k ate quantVertices
  for i ate quantVertices
    for j ate quantVertices
      se (dist[i][k] + dist[k][j]) < dist[i][j]
        dist[i][j] = dist[i][k] + dist[k][j]
```

**Dijkstra(partida, destino, quantVertices)**

```
CriaVetoresAux(quantVertices)      // Cria vetores de distancia, pai, conjuntos Q e S
Inicializa(partida, quantVertices)  // Inicializa os vetores auxiliares criados
enquanto a menor distancia definitiva do destino não for conhecida
    u ← extrairMinimo(quantVertices)
    for cada v ∈ Adj[u]
        Relaxa(u, v, w)
```

**Inicializa(partida, quantVertices)**

```
for cada v ∈ V
    d[v] ← ∞
    π[v] ← NULL
d[s] ← 0
```

**Relaxa(u, v, w)**

```
se d[v] > (d[u] + w(u,v)) então
    d[v] ← d[u] + w(u, v)
    π[v] ← u
```

## Saídas encontradas

```
D:\Aeds3\Trabalho2>main n25.txt
n25.txt
23 20 13 4 12
peso = 36

D:\Aeds3\Trabalho2>main n50.txt
n50.txt
27 6 0 20
peso = 32

D:\Aeds3\Trabalho2>main n100.txt
n100.txt
72 90 8 50 24 56 48 4
peso = 46

D:\Aeds3\Trabalho2>main n250.txt
n250.txt
122 152 89 220 77 1 102 13
peso = 52

D:\Aeds3\Trabalho2>main n500.txt
n500.txt
277 201 58 158 283 239 43
peso = 63

D:\Aeds3\Trabalho2>main n750.txt
n750.txt
369 124 2 530 227 529 193
peso = 70

D:\Aeds3\Trabalho2>main n1000.txt
n1000.txt
900 78 58 458 987 325 922 181 221
peso = 68
```

## Conclusão

Iniciamos o trabalho armazenando os dados em um vetor de struct e tentando executar Dijkstra a partir dele. Isso gerou problemas e deixou sua execução muito lenta. A partir desse momento, fomos aconselhados a utilizar o algoritmo de Floyd Warshall, e para implementá-lo, tivemos que mudar por completo a representação dos dados. Passamos a representar os dados em forma de matriz. Com isso, conseguimos encontrar os dois vértices mais distantes do grafo e seu peso. Contudo, essa resposta não gerava todos os vértices desse caminho. Depois de descobrir os dois vértices, executamos o algoritmo de Dijkstra até encontrar a distância definitiva entre eles. Assim, foi possível resgatar todos os vértices desse caminho através do vetor  $\pi$  (que armazena o pai de cada vértice).