

Caminho mais curto

Num grafo **pesado**, com pesos **w**, o **peso do caminho**

$$p = v_0 v_1 \dots v_k$$

é a **soma dos pesos dos arcos** que o integram

$$w(p) = \sum_{i=1}^k w(v_{i-1}, v_i)$$

O caminho **p** é **mais curto** que o caminho **p'** se o **peso** de **p** é **menor** que o **peso** de **p'**

Cálculo dos caminhos mais curtos

3 algoritmos

1. Cálculo dos caminhos mais curtos num grafo orientado acíclico (DAG), com pesos possivelmente negativos
2. Algoritmo de Dijkstra, só aplicável a grafos sem pesos negativos
3. Algoritmo de Bellman-Ford, aplicável a qualquer grafo pesado

Estes algoritmos calculam os caminhos mais curtos de um nó **s** para os restantes nós do grafo (*single-source shortest paths*)

Caminhos mais curtos

Funções comuns aos diversos algoritmos

O peso do caminho mais curto de **s** a qualquer outro nó é inicializado com ∞

INITIALIZE-SINGLE-SOURCE(*G*, *s*)

```
1 for each vertex v in G.V do
2   v.d ← INFINITY    // peso do caminho mais curto de s a v
3   v.p ← NIL         // predecessor de v nesse caminho
4 s.d ← 0
```

Se o caminho de **s** a **v**, que passa por **u** e pelo arco **(u, v)**, tem menor peso do que o mais curto anteriormente encontrado, encontrámos um caminho mais curto

RELAX(*u*, *v*, *w*)

```
1 if u.d + w(u,v) < v.d then
2   v.d ← u.d + w(u,v)
3   v.p ← u
```

Caminhos mais curtos a partir de um vértice

Algoritmo para DAGs

$G = (V, E)$ – DAG pesado (pode ter pesos negativos)

DAG-SHORTEST-PATHS(G, w, s)

```
1 topologically sort the vertices of G
2 INITIALIZE-SINGLE-SOURCE( $G, s$ )
3 for each vertex  $u$ , taken in topologically sorted
                                     order do
4     for each vertex  $v$  in  $G.\text{adj}[u]$  do
5         RELAX( $u, v, w$ )
```

Caminhos mais curtos a partir de um vértice

Algoritmo de Dijkstra

$G = (V, E)$ – grafo pesado orientado (sem pesos negativos)

DIJKSTRA(G, w, s)

```
1 INITIALIZE-SINGLE-SOURCE( $G, s$ )
2  $Q \leftarrow G.V$  // fila com prioridade, chave  $u.d$ 
3 while  $Q \neq \text{EMPTY}$  do
4      $u \leftarrow \text{EXTRACT-MIN}(Q)$ 
5     for each vertex  $v$  in  $G.\text{adj}[u]$  do
6         RELAX( $u, v, w$ ) // pode alterar  $Q$ !
```

Quando é encontrado um novo caminho mais curto para um vértice (na função RELAX), é necessário reorganizar a fila Q (DECREASE-KEY)

Caminhos mais curtos a partir de um vértice

Algoritmo de Bellman-Ford

$G = (V, E)$ – grafo pesado orientado (pode ter pesos negativos)

BELLMAN-FORD(G, w, s)

```
1 INITIALIZE-SINGLE-SOURCE( $G, s$ )
2 for  $i \leftarrow 1$  to  $|G.V| - 1$  do
3     for each edge  $(u,v)$  in  $G.E$  do
4         RELAX( $u, v, w$ )
5 for each edge  $(u,v)$  in  $G.E$  do
6     if  $u.d + w(u,v) < v.d$  then
7         return FALSE
8 return TRUE
```