Aula 8

- Fecho Transitivo
- Repesagem de Arcos
- Algoritmo de Johnson
  - Complexidade
  - Correcção
- Problemas

## - Caminhos mais curtos entre todos os pares

- Queremos calcular duas matrizes: $\Pi$ e $D$.
  - $\Pi_{ij}$: predecessor de $j$ no caminho mais curto entre $i$ e $j$ se $j$ é atingível a partir de $i$
  - $D_{ij}$: distância do caminho mais curto entre $i$ e $j$.

- $G_{\Pi,i} = (V_{\Pi,i}, E_{\Pi,i})$ // O grafo dos caminhos mais curtos com origem em $i$

$$E_{\Pi,i} = \{ (\Pi_{ij}, j) \mid \Pi_{ij} \neq Nil \}$$
$$V_{\Pi,i} = \{ j \mid \Pi_{ij} \neq Nil \} \cup \{ i \}$$

### Algoritmo 3 [Caminhos Mais Curtos entre todos os pares - Solução Recursiva]

$$l_{ij}^{(0)} = \begin{cases} 0 & \text{se } i = j \\ \infty & \text{c.c.} \end{cases}$$

$$l_{ij}^{(m)} = \min \left( l_{ij}^{(m-1)}, \min \{ l_{ik}^{(m-1)} + w_{kj} \mid 1 \leq k \leq n \} \right)$$
$$= \min \{ l_{ik}^{(m-1)} + w_{kj} \mid 1 \leq k \leq n \}$$

- UpdateAllPairsShortestPaths(G, D)

```
let D' be a new matrix with the dimensions of D
for i = 1 to G.n
    for j = 1 to G.n
        D'ij = ∞
        for k = 1 to G.n
            D'ij = min(D'ij, Dik + G.Wkj)
return D'
```

[Complexidade]
$O(V^3)$

Nota: No CLRS os autores alocam a matriz duas vezes (páginas 688 e 689)

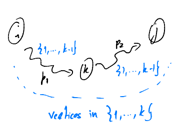
- ComputeAllPairsShortestPaths(G)

```
let D = G.W
for m = 2 to |G.V| - 1
    D = UpdateAllPairsShortestPaths(G, D)
```

[Complexidade]
$O(V^4)$

### Algoritmo 4 [Floyd Warshall Algorithm]

$$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{se } k = 0 \\ \min \left( d_{ik}^{(k-1)} + d_{kj}^{(k-1)}, d_{ij}^{(k-1)} \right) & \text{c.c.} \end{cases}$$




FloydWarshall(G)

```
let D' = G.W
for k = 2 to n
    D = D'; D' = NewMatrix(G.n, G.n)
    for i = 1 to n
        for j = 1 to n
            Dij = min(Dij, Dik + Dkj)
return D'
```

Remark:
- Podemos deixar de usar D' e usar apenas D?
- É a mesma coisa?

Floyd Warshall$(G)$

let $D^{(1)} = G.W$

let $\Pi = initPi(G.W)$

for $k = 2$ to $n$

  let $D^{(k)}$ be a new matrix with the same dimensions as $D^{(1)}$

  for $i = 1$ to $n$

    for $j = 1$ to $n$

      $d = D_{ik} + D_{kj}$

      if $d < D_{ij}$

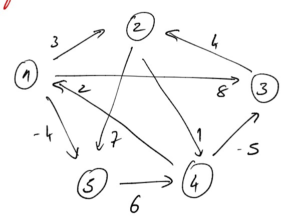
        $D_{ij}^{(k)} = d$; $\Pi_{ij} = \Pi_{kj}$

$$\pi_{ij}^{(0)} = \begin{cases} i & \text{se } (i,j) \in E \\ nil & \text{c.c.} \end{cases}$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{se } d_{ij}^{k} = d_{ij}^{(k-1)} \\ \pi_{kj}^{(k-1)} & \text{se } d_{ij}^{k} < d_{ij}^{(k-1)} \end{cases}$$

Complexidade: $\Theta(V^3)$

Exemplo 3 [Caminhos Mais Curtos entre Todos os Pares]



$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

$$D^{0} = \begin{array}{c|c} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} & \\ \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \end{array}$$

O vértice 4 é o único q consegue chegar a 1 por isso é o único cujas estimativas podem ser melhoradas.

$d_{41} + d_{1j}$

$$D^{1} = \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$

$$D^{3} = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$

$$D^{2} = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix}$$

$$D^{4} = \begin{bmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix}$$

$$D^{5} = \begin{bmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{bmatrix}$$

Definição 1 [Fecho Transitivo]

Seja $G=(V,E)$ um grafo dirigido, o fecho transitivo de $G$, é definido como se segue:

$$G^* = (V, E^*) \quad \text{onde} \quad E^* = \{(u,v) \mid u,v \in V \wedge u \rightsquigarrow v\}$$

Algoritmo 1 [Floyd Warshall p/ fecho transitivo]

$$t_{ij}^{(0)} = \begin{cases} \text{true} & \text{se } i=j \vee (i,j) \in E \\ \text{false} & \text{c.c.} \end{cases}$$

$$t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee \left(t_{ik}^{(k-1)} \vee t_{kj}^{(k-1)}\right)$$

```
TransitiveClosure(G)
  let T' = InitTMatrix(G)
  for k=1 to n
    T = T'; T' = NewMatrix(G.n, G.n)
    for i=1 to n
      for j=1 to n
        T'ij = Tij ∨ (Tik ∧ Tkj)
  return T'
```

· Algoritmo de Johnson

Ideia:

- Dado um grafo pesado $G$ c/ arcos c/ pesos negativos, calcular um grafo $G'$ tal q os caminhos mais curtos em $G$ mas $G'$ não tem arcos c/ pesos negativos.
- Aplicar o algoritmo de Dijkstra a cada vértice de $G'$ para calcular os caminhos mais curtos c/ origem nesse vértice.

Lema 1 [Repesagem dos Arcos]

Seja $G=(V,E,w)$ um grafo dirigido pesado e $h: V \longrightarrow \mathbb{R}$ uma função e seja $\hat{w}: E \longrightarrow \mathbb{R}$ a seguinte função:

$$\hat{w}(u,v) = w(u,v) + h(u) - h(v)$$

Então:

1. se $p$ é um caminho mais curto no grafo $G=(V,E,w)$, então $p$ tb é um caminho mais curto em $\hat{G}=(V,E,\hat{w})$.
2. $G$ contém ciclos negativos sse $\hat{G}$ contém ciclos negativos

**Prova:**

① $p$ é um caminho mais curto em $G$ $\iff$ $p$ é um caminho mais curto em $\hat{G}$

· Seja $p = \langle v_0, \dots, v_n \rangle$, começamos por mostrar que

$$\hat{w}(p) = w(p) + h(v_0) - h(v_n)$$

$$0 \xrightarrow{w_{01}} 1 \xrightarrow{w_{12}} 2 \xrightarrow{w_{23}} 3$$

$$w(p) = (w_{01} + h(v_0) - h(v_1)) + (w_{12} + h(v_1) - h(v_2)) + (w_{23} + h(v_2) - h(v_3))$$
$$= (w_{01} + w_{12} + w_{13}) + h(v_0) + (h(v_1) - h(v_1)) + (h(v_2) - h(v_2)) - h(v_3)$$
$$= w(p) + h(v_0) - h(v_3)$$

$$- \hat{w}(p) = \sum_{i=0}^{n-1} \hat{w}(v_i, v_{i+1})$$
$$= \sum_{i=0}^{n-1} \left( w(v_i, v_{i+1}) + h(v_i) - h(v_{i+1}) \right)$$
$$= \sum_{i=0}^{n-1} w(v_i, v_{i+1}) + \sum_{i=0}^{n-1} h(v_i) - \sum_{i=0}^{n-1} h(v_{i+1})$$
$$= \sum_{i=0}^{n-1} w(v_i, v_{i+1}) + \cancel{\sum_{i=1}^{n-1} h(v_i)} + h(v_0) - \left( \cancel{\sum_{i=1}^{n-1} h(v_i)} + h(v_n) \right)$$
$$= \sum_{i=0}^{n-1} w(v_i, v_{i+1}) + h(v_0) - h(v_n)$$

· $\hat{w}(p) = w(p) + \underbrace{h(v_0) - h(v_n)}$

Esta quantidade não depende do caminho $p$ que liga $v_0$ a $v_n$

Assim sendo, o caminho mais curto entre quaisquer dois vértices $v_0$ e $v_n$ em $G$ coincide com o caminho mais curto entre esses mesmos dois vértices em $\hat{G}$.

② Suponhamos que $p = \langle v_0, \dots, v_n \rangle$ é um caminho circular em $G$ e que portanto $v_0 = v_n$.

Segue que $\hat{w}(p) = w(p) + h(v_0) - h(v_n)$
$= w(p) + h(v_0) - h(v_0)$
$= w(p)$

Concluímos que o peso dos caminhos circulares não é alterado pela função de repesagem.

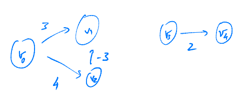
De onde segue que um caminho circular tem um peso negativo em $G$ sse tb tem peso negativo em $\hat{G}$.

Observação: Dado um grafo pesado dirigido $G = (V, E, w)$, queremos encontrar uma função de repesagem $h$ tal que:

$$\forall (u,v) \in E.\ \hat{w}(u,v) \geq 0$$
$$\iff w(u,v) + h(u) - h(v) \geq 0$$
$$\iff \underbrace{h(v) \leq h(u) + w(u,v)}_{\text{Desigualdade Triangular}}$$

· Dado um grafo $G = (V, E)$, escolhemos um vértice $s$ e calculamos p/ todo o $v \in V$ $\delta(s, v)$, usando o algoritmo de Johnson.

Problema: vértices não atingíveis $(\underbrace{w + \infty - \infty}_{?})$




$h(v_1) = -1 \quad h(v_2) = 4 \quad h(v_3) = \infty \quad h(v_4) = \infty \quad h(v_0) = 0$

$$\hat{w}(v_3, v_4) = w(v_3, v_4) + h(v_3) - h(v_4)$$
$$= 2 + \infty - \infty$$
$$= ?$$

**Definição 2 [Grafo Estendido c/ Vértice de Origem]**

Seja $G = (V, E, w)$ um grafo dirigido pesado, a extensão de $G$ ao vértice $s$, denotada por $\bar{G}_s = (\bar{V}, \bar{E}, \bar{w})$ é definida como se segue:

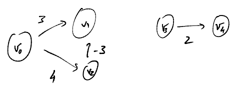
- $\bar{V} = V \cup \{s\}$
- $\bar{E} = E \cup \{(s, v) \mid v \in V\}$
- $\bar{w}(v_1, v_2) = \begin{cases} w(v_1, v_2) & \text{if } v_1, v_2 \neq s \\ 0 & \text{if } v_1 = s \land v_2 \in V \end{cases}$

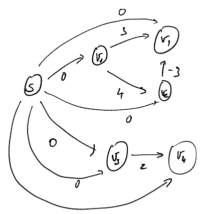
## Definição 3 [Altura de Johnson]

Seja $G = (V, E, w)$ um grafo pesado dirigido, a função de altura $h: V \to \mathbb{R}$ de Johnson é definida como se segue:

$$h(v) = \delta_{G_s}(s, v) \quad , \text{ onde } s \notin V$$

## Exemplo 1 [Altura de Johnson]



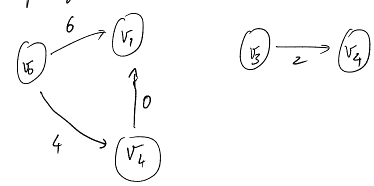




$h(v_0) = 0$ $\quad$ $h(v_3) = 0$

$h(v_1) = -3$ $\quad$ $h(v_4) = 0$

$h(v_2) = 0$

Grafo repesado:




## Algoritmo 2 [Johnson]

- Calcular os caminhos curtos entre todos os pares em $G = (V, E, w)$

① Calcular o grafo estendido $\bar{G}_s = (\bar{V}, \bar{E}, \bar{w})$ com $s \notin V$ $\quad O(V+E)$

② Usar o algoritmo de Bellman-Ford p/ determinar a função de altura $h$
Se o algoritmo de Bellman-Ford retornar falso, o algoritmo de Johnson tb retorna falso. $\quad \| O(E \cdot V)$

③ Calcular o grafo repesado $\hat{G} = (V, \hat{E}, \hat{w})$ $\quad \| O(V+E)$

④ Para cada vértice $u \in V$, usar o algoritmo de Dijkstra $D_{uv}$ e $\Pi_{uv}$ para todo $v \in V$ $\quad \| O(E \lg V)$ $\quad | \; O(V \cdot E \cdot \lg V)$

⑤ Retornar $D$ e $\Pi$.

[Complexidade]

$O(V E \lg V)$