
Grafos: caminhos (matriz adjacência)

Algoritmos e Estruturas de Dados 2

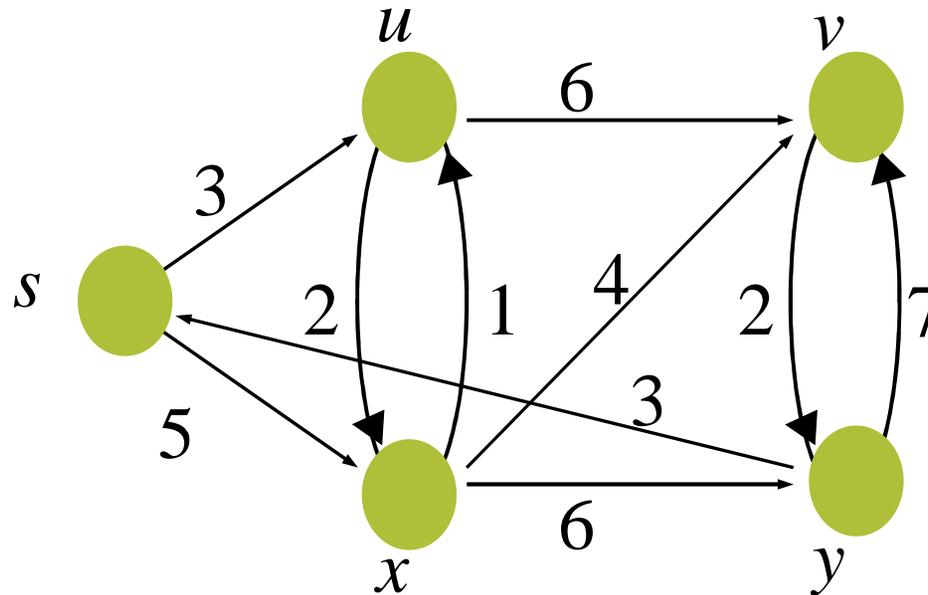
Graça Nunes

O problema do menor caminho

- Um motorista deseja encontrar o caminho mais curto possível entre duas cidades do Brasil
- Caso ele receba um mapa das estradas de rodagem do Brasil, no qual a distância entre cada par adjacente de cidades está exposta, como poderíamos determinar uma rota mais curta entre as cidades desejadas?
- Uma maneira possível é enumerar todas as rotas possíveis que levam de uma cidade à outra, e então selecionar a menor

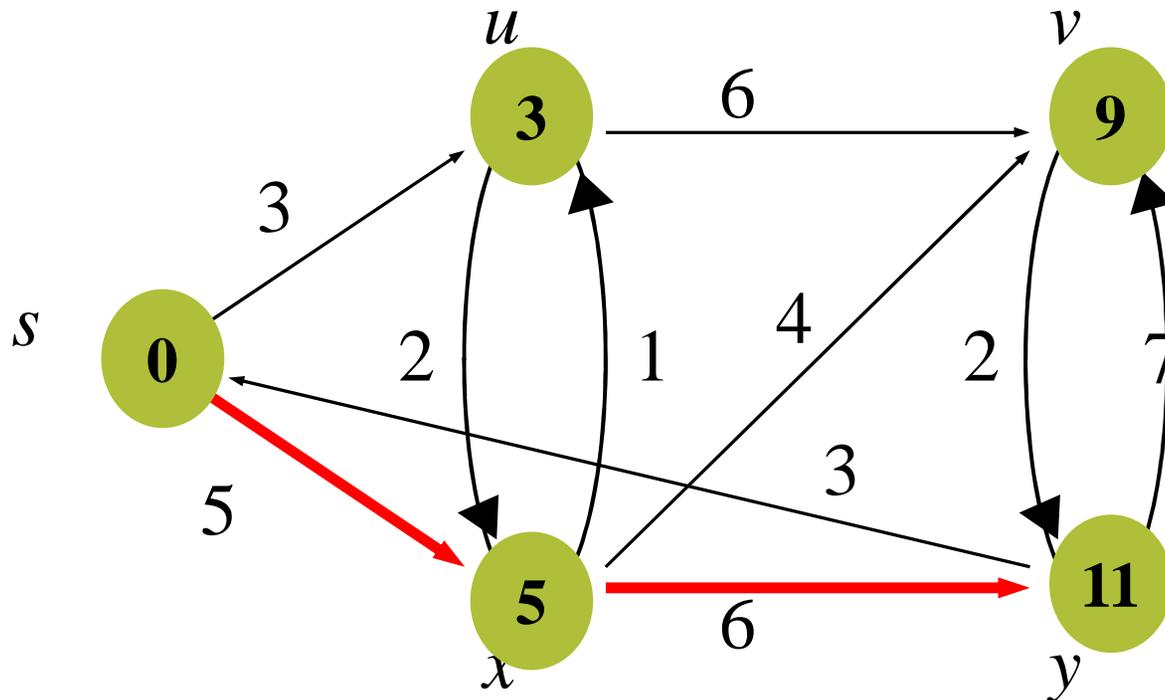
Caminhos mínimos

- O problema do caminho mínimo consiste em determinar um menor caminho entre um vértice de origem u e um vértice de destino v
- Qual o menor caminho entre s e y ?



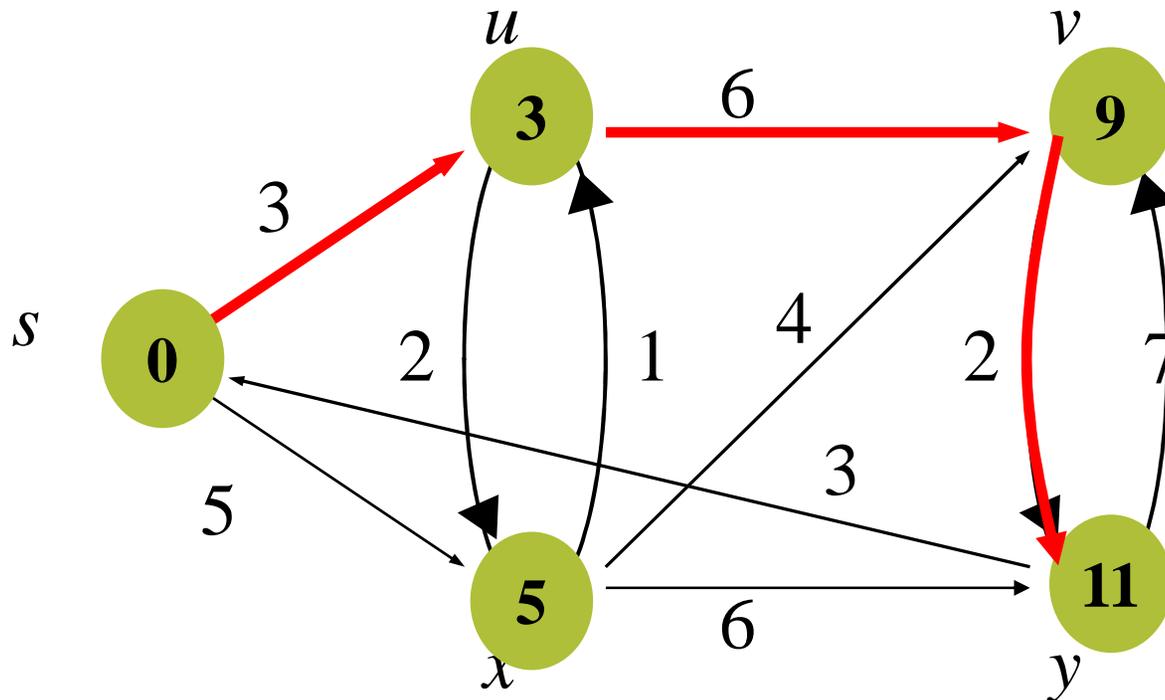
Caminhos mínimos

- Possíveis caminhos simples



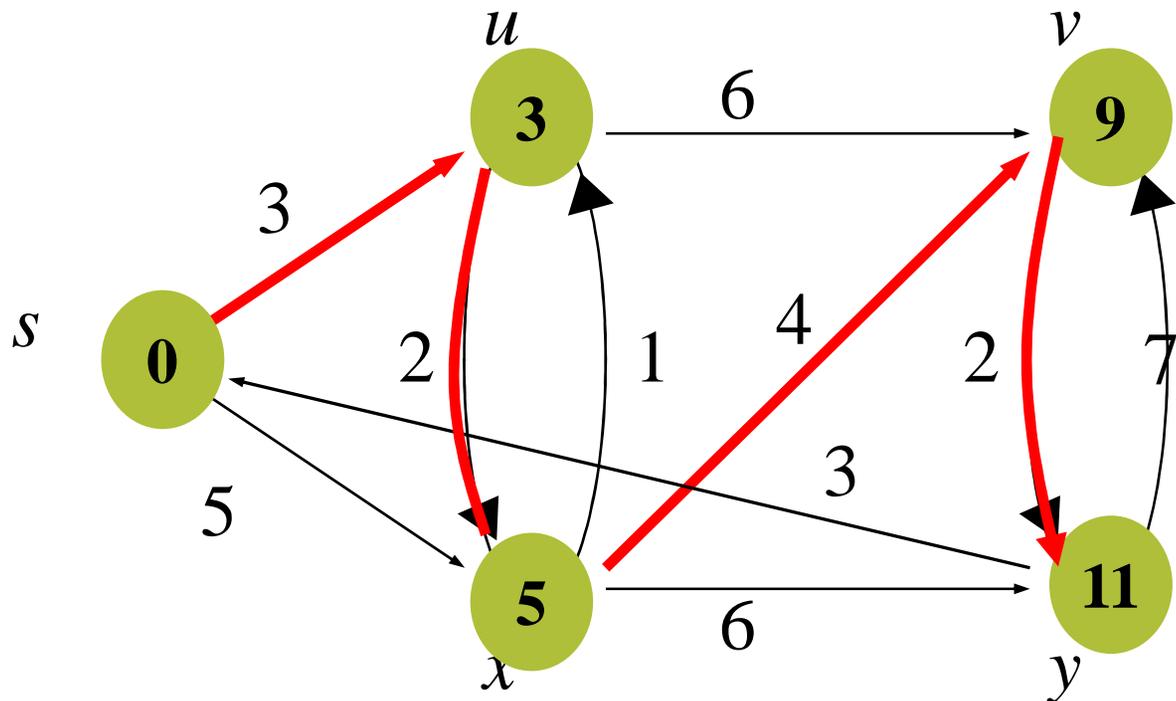
Caminhos mínimos

- Possíveis caminhos simples



Caminhos mínimos

- Possíveis caminhos simples



Além de outros com ciclos....

Caminho mínimo

- **Duas abordagens** para caminho mínimo
 - Se grafo **não valorado** (assume-se que cada aresta tem peso 1), busca em largura é uma boa opção → o caminho mínimo coincide com o caminho mais curto
 - Se grafo **valorado**, outros algoritmos são necessários

Caminho mínimo

- Grafo dirigido $G(V,E)$ com função peso $w: E \rightarrow \mathfrak{R}$ que mapeia as arestas em pesos
- Peso (custo) do caminho $p = \langle v_0, v_1, \dots, v_k \rangle$

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

- Caminho de menor peso entre u e v :

$$\delta(u,v) = \begin{cases} \min\{w(p) : u \xRightarrow{p} v\} & \text{se } \exists \text{ rota de } u \text{ p/ } v \\ \infty & \text{cc} \end{cases}$$

Caminho mínimo

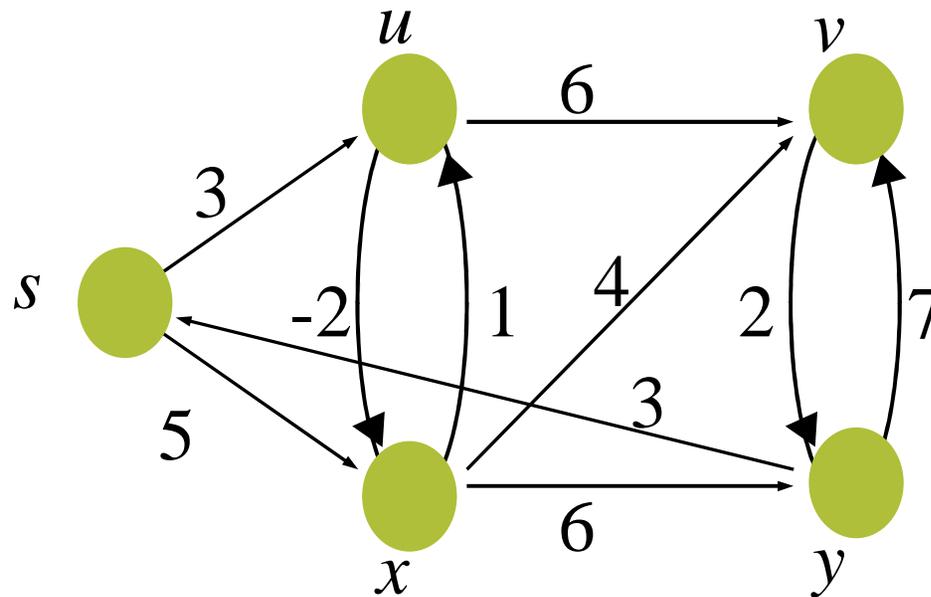
- Caminho mínimo entre os vértices u e v definido como qualquer rota p com um peso

$$w(p) = \delta(u, v)$$

- Atenção especial com ciclos e pesos negativos

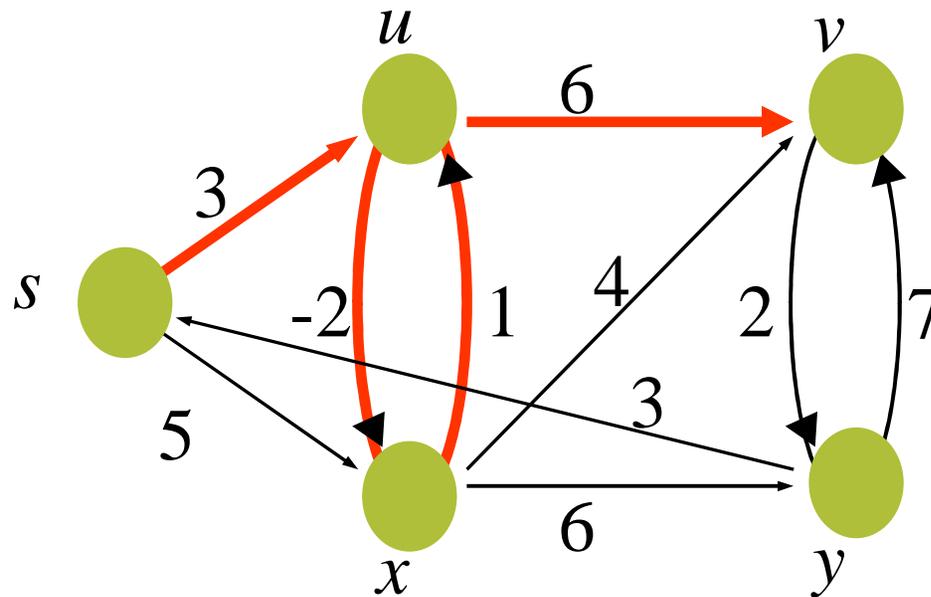
Caminho mínimo

- Qual o caminho mínimo entre s e v ?



Caminho mínimo

- Qual o caminho mínimo entre s e v ?



Caminho mínimo

- Se há um ciclo positivo no caminho, ele não faz parte do caminho mínimo
 - Por quê?

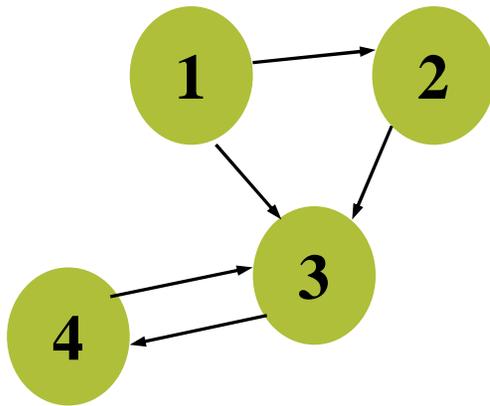
Algoritmos para Dígrafos representados em Matrizes de Adjacência

- Perguntas possíveis de se responder para um dígrafo com n vértices e m arestas:
 - **P1:** Existe caminho de v_i a v_j ? (valorados ou não)
 - **P2:** Qual o número de caminhos de comprimento k de v_i a v_j ? (não valorados)
 - **P3:** Qual é o comprimento do caminho mínimo de v_i a v_j ? (valorados)
 - **P4:** Qual é o custo do caminho mínimo de v_i a v_j ? (valorados)
 - **P5:** Qual é o caminho mínimo (menor custo) de v_i a v_j ? (valorados)

Matriz Caminho - P

- Def.: A Matriz Caminho, P, de um dígrafo $D(V,A)$ é definida como:
 - $p_{ij} = 1 \iff \exists$ um caminho de v_i para v_j
 - $p_{ij} = 0$, c.c.

■ Ex.



X: Matriz Adjacência

P: Matriz Caminho

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Obs.: $p_{ii} = 1 \iff \exists$ um ciclo a partir de v_i

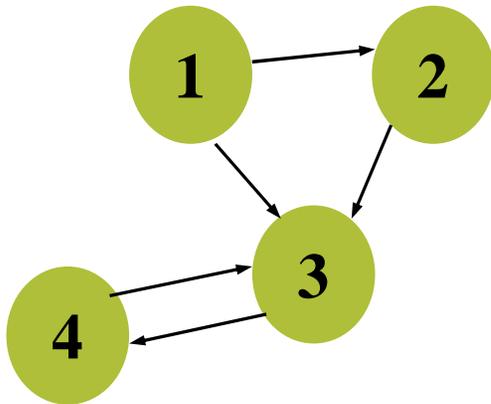
-
- Precisamos então de um algoritmo para calcular P , a partir de X , para então responder à pergunta:
 - **P1:** Existe caminho de v_i a v_j ? (valorados ou não)

(adiante....)

Teorema

- Seja X a matriz adjacência de D , e seja $Y=X^h$. Então y_{ij} é o número total de sequências distintas $(v_i, \dots), \dots, (\dots, v_j)$ que
 - Possuem comprimento h
 - Correspondem a caminhos em D
- Exercício: provar por indução sobre h

No exemplo anterior



X: Matriz Adjacência

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

X^2

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

X^3

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$X^4 = X^2$

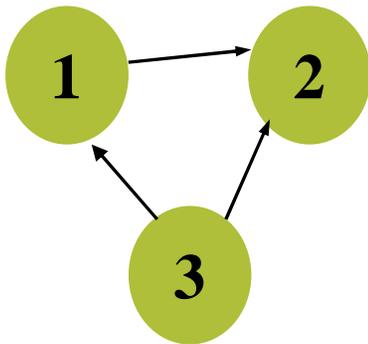
$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$X^5 = X^3$

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Corolário 1:

- Se $X^h =$ Matriz Nula, para algum $h \leq n$, então D é acíclico.
- Verifique para o exemplo ($n = 3$):



-
- Já temos então um algoritmo que calcula X^h , de ordem n^3 , para responder à pergunta:
 - **P2**: Qual o número de caminhos de comprimento k de v_i a v_j ? (não valorados)

Corolário 2:

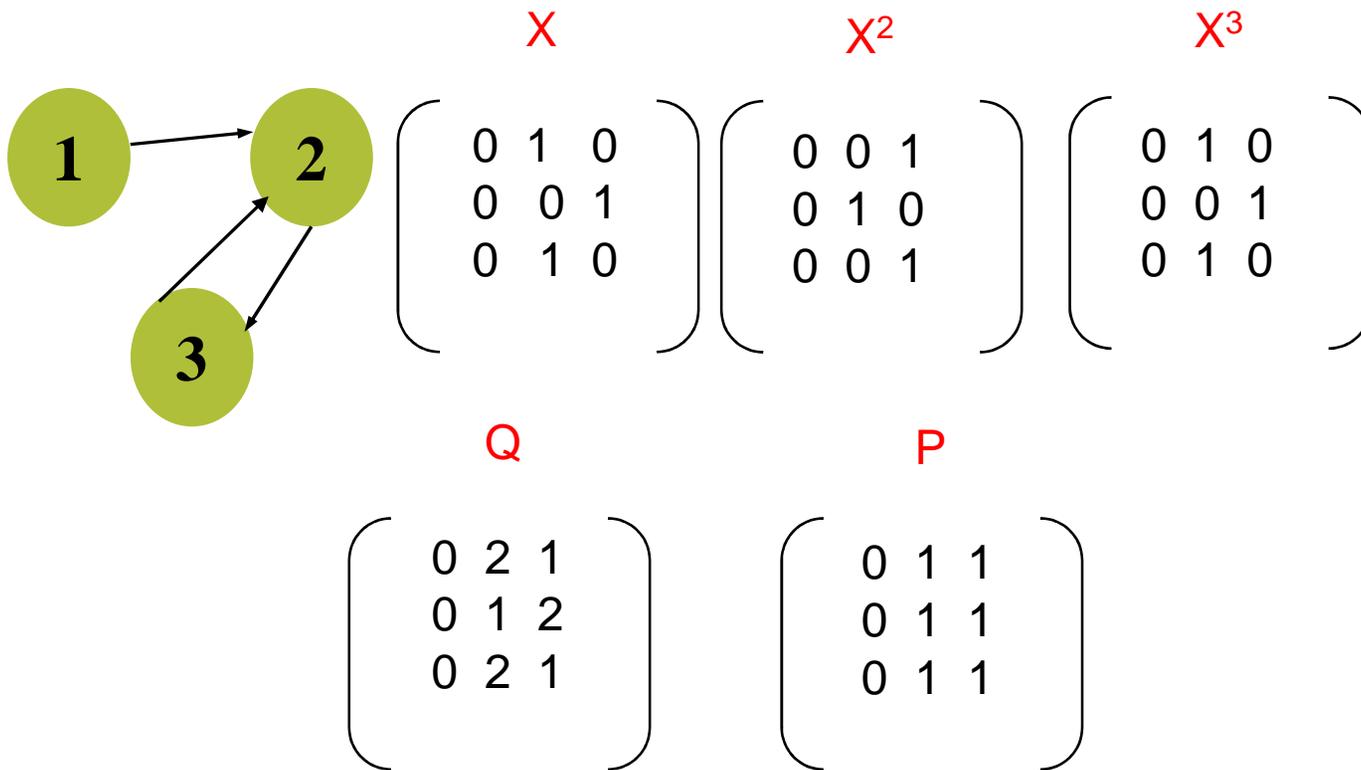
- Se X é matriz adjacência de D e $Q = X + X^2 + X^3 + \dots + X^n$, então a matriz caminho P é tal que

$$p_{ij} = 1 \iff q_{ij} \neq 0$$

Repare que:

- q_{ij} é a quantidade de caminhos possíveis de v_i a v_j , de comprimentos $\leq n$ (pode haver mais, sse D for cíclico)
- Pelo Corolário 1, se há caminho de v_i a v_j , ele aparecerá até a potência n de X

Exemplo



-
- O Corolário 2 nos dá um algoritmo para calcular a matriz caminho P a partir de X , mas ele é muito caro, visto que envolve muitos produtos de matrizes.
 - Para torná-lo mais eficiente, usamos operações booleanas **and** \wedge e **or** \vee no lugar de multiplicação e soma, e redefinimos X^k e Q , a partir de X , com valores booleanos 0 e 1.

$$X^k : \quad x_{ij}^k = \bigvee_{k=1}^n (x_{ik}^{k-1} \wedge x_{kj}) ; 1 \leq i, j \leq n$$

$$Q : \quad q_{ij} = x_{ij} \vee x_{ij}^2 \vee \dots \vee x_{ij}^n$$

Mas então Q é a própria matriz P desejada!

Então $P = X + X^2 + X^3 + \dots + X^n$,

onde $X^k = X^{k-1} \wedge X$

Algoritmo de Roy-Warshall (RW)

- Calcula P, a matriz caminho de D, a partir de sua matriz adjacência, X.

Algoritmo:

Dada $X_{n \times n}$, matriz adjacência de $D(V,A)$

Início

Faça $P = X$;

para $j = 1$ até n faça

para $i = 1$ até n faça

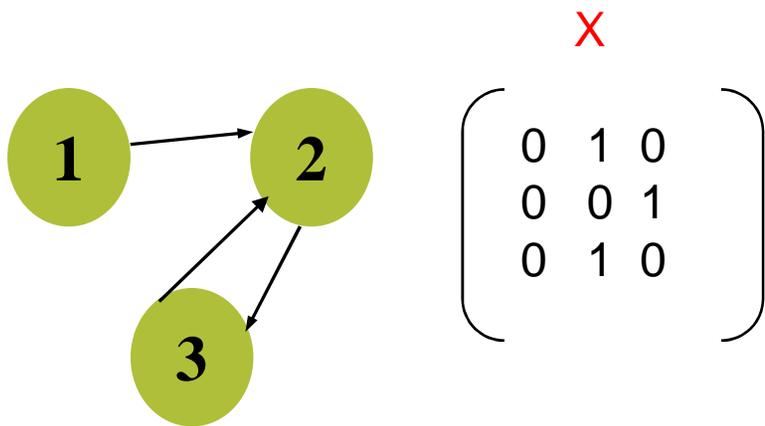
se $p_{ij} = 1$ /*se já há caminho de i a j */

então para $k = 1$ até n faça

$p_{ik} = p_{ik} \vee p_{jk}$ /*então haverá de
 i a k se houver de j a k */

Fim

Exemplo



Então P1 (Existe caminho de v_i a v_j ?) já pode ser respondida

Para dígrafos valorados: calculando o caminho mínimo entre 2 vértices

- Neste caso, X , a matriz adjacência, tem os valores 1 substituídos pelos respectivos pesos.
- Seja MC , a **Matriz dos custos dos caminhos mínimos**, tal que:

$$MC_{ij} = \delta(i,j)$$
$$\delta(u,v) = \begin{cases} \min\{w(p) : u \xrightarrow{p} v\} & \text{se } \exists \text{ rota de } u \text{ para } v \\ \infty & \text{cc} \end{cases}$$

Ou seja, MC_{ij} é o custo do caminho mínimo entre i e j .

- ~~■ Estratégia: Trocar 0 por ∞ e 1 pelo peso, em X~~

Algoritmo de Floyd-Warshall(FW)

Dada X de D, cria MC, a matriz dos custos dos caminhos mínimos

Início

faça MC = X

para j = 1 até n faça

para i = 1 até n faça

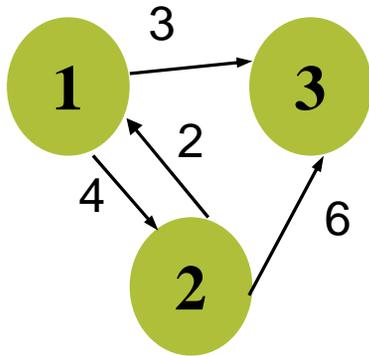
se $MC_{ij} \neq \infty$ então /*se há caminho de custo MC_{ij} de i a j*/

para k = 1 até n faça

$MC_{ik} = \min (MC_{ik}, MC_{ij} + MC_{jk})$ /*então o custo min. de i a k é o mínimo entre o caminho direto de i a k e a soma de i a j e de j a k*/

Fim

Exemplo



$$\begin{array}{c}
 \text{X = MC}_1 \\
 \left(\begin{array}{ccc}
 \infty & 4 & 3 \\
 2 & \infty & 6 \\
 \infty & \infty & \infty
 \end{array} \right) \sim \begin{array}{c}
 \text{MC} \\
 \left(\begin{array}{ccc}
 (\infty, 4+2) & (4, \infty) & \\
 6 & 4 & 3 \\
 (2, \infty) & 2 & 6 & 5 \\
 \infty & (\infty, 2+4) & \infty & \infty
 \end{array} \right)
 \end{array}
 \end{array}$$

Additional annotations in blue: $(3, 4+6)$ next to the top-right cell of the second matrix, and $(6, 2+3)$ next to the middle-right cell of the second matrix.

E assim RF responde P4: Qual é o custo do caminho mínimo de v_i a v_j ? (valorados)

Consequência

- O que acontece se, no algoritmo FW, todo peso for =1?

MC_{ij} é o comprimento do menor caminho entre v_i e v_j

Assim, RF também responde:

P3: Qual é o comprimento do caminho mínimo de v_i a v_j ? (valorados)

Queremos mais...

- Além do custo do caminho mínimo, e do seu comprimento, queremos o próprio caminho, ou seja, a sequência de vértices que o compõe
- Basta uma pequena alteração no algoritmo FW:
- Calcula-se simultaneamente a MC, uma matriz M , de igual dimensão de X .

Cálculo do caminho mínimo

- No início:

$$M_{ik} = k, \text{ se } X_{ik} \neq \infty$$

$$M_{ik} = 0, \text{ se } X_{ik} = \infty$$

- Ou seja, no início, M_{ik} representa o vértice onde incide a aresta que parte de v_i (0 indica inexistência de aresta)
- A ideia é alterar os valores de M sempre que um caminho alternativo é calculado (i.e. quando o custo mínimo é alterado), de forma que no final, M_{ik} contém o rótulo do próximo vértice do caminho mínimo de v_i a v_k

Cálculo do caminho mínimo

- P.ex.
- Se $(v_i, v_t, v_u, \dots, v_k)$ é o cam. mínimo de v_i a v_k , então $M_{ik} = t$
- Mas o restante do caminho é facilmente recuperado, uma vez que M_{tk} tem o próximo vértice do caminho: v_u
- E assim por diante

Algoritmo FW com especificação de caminho – FW+

Dada X, cria MC e M

Início

faça MC = X

inicialize M conforme a definição*

para j = 1 até n faça

 para i = 1 até n faça

 se $MC_{ij} \neq \infty$ então

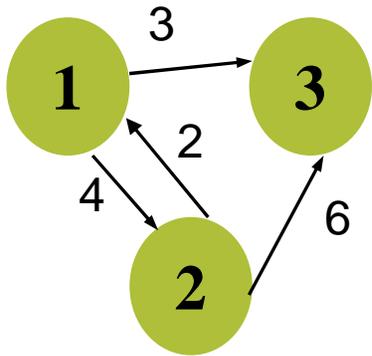
 para k = 1 até n faça

 se $MC_{ik} > MC_{ij} + MC_{jk}$ então

 { $MC_{ik} = MC_{ij} + MC_{jk};$
 $M_{ik} = M_{ij}$ }

Fim

Calculando M para o exemplo anterior

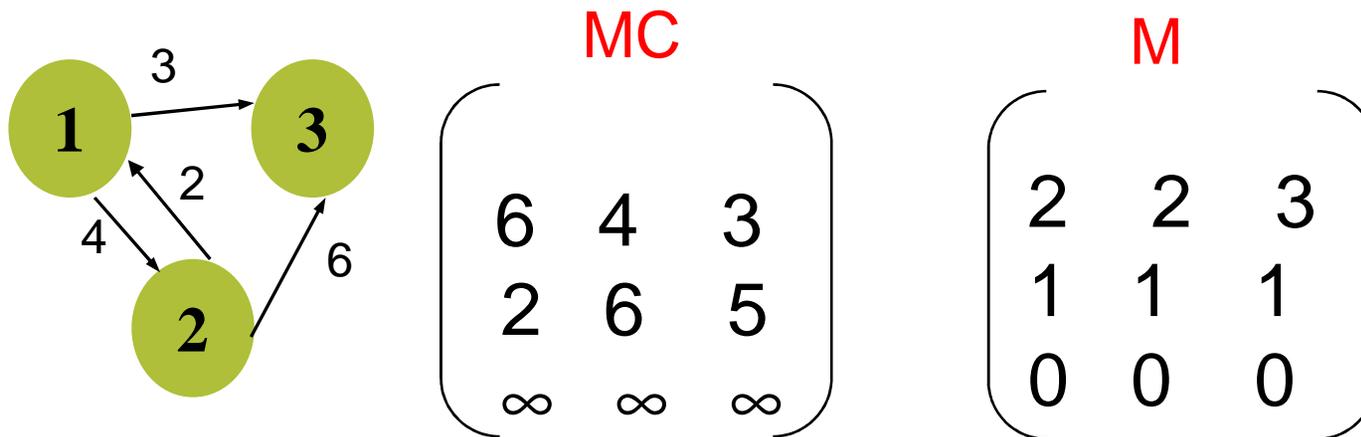


$$X \begin{pmatrix} \infty & 4 & 3 \\ 2 & \infty & 6 \\ \infty & \infty & \infty \end{pmatrix}$$

$$M_1 \begin{pmatrix} 0 & 2 & 3 \\ 1 & 0 & 3 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\sim \begin{pmatrix} 2 & 2 & 3 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix} M$$

Calculando M para o exemplo anterior



Leitura das Matrizes M e MC:

- o c.m. de v1 a v1 é (v1,v2,v1) com custo 6
- o c.m. de v1 a v2 é (v1,v2) com custo 4
- o c.m. de v1 a v3 é (v1,v3) com custo 3
- o c.m. de v2 a v1 é (v2,v1) com custo 2
- o c.m. de v2 a v2 é (v2,v1,v2) com custo 6
- o c.m. de v2 a v3 é (v2,v1,v3) com custo 5
- Não existem caminhos a partir de v3

Concluindo

- E finalmente FW+ responde também
P5: Qual é o caminho mínimo (menor custo) de v_i a v_j ? (valorados)
- Repare que, ao usar os algoritmos sobre a Matriz de Adjacência, temos as respostas simultaneamente para todo par de vértices (i,j) , a um custo de $O(n^3)$
- A seguir, veremos algoritmos sobre Listas de Adjacências, que respondem às perguntas, dados (i,j) específicos.

Exercícios

- Programe os algoritmos vistos usando o TAD Matriz de Adjacência