# SCC0216 - Modelagem Computacional em Grafos

## Caminhos Mínimos

Prof. Alneu (alneu@icmc.usp.br ) / Profa. Rosane (rminghim@icmc.usp.br)

PAE: Alan (alan@icmc.usp.br) / Henry (henry@icmc.usp.br)

Baseado no material de aula original: Profª. Josiane M. Bueno

# Caminho mínimo

- **Problema: encontrar o caminho de menor custo (ou o menor caminho) entre dois vértices em um grafo valorado**
  - ***Algoritmo de Djikstra***
  - ***Algoritmo de Floyd-Warshall***

# Caminho mínimo

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

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

  - Caminho de menor peso entre *u* e *v*:

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

# Caminho mínimo

- Menor caminho entre os vértices *u* e *v* definido como qualquer rota *p* com um peso:

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

# Dijkstra($L, v_p, v_q$)

-$T=\{v_p\}$ ; $PL=\{0\}$; $P=\{0\}$; $W=V-T=V-\{v_p\}$ //inicialização

-$TL=\{\infty$ qq $w_i$ pertencente $W\}$

-Enquanto **$v_q \notin T$** OU **$W \neq Ø$**

-Determine **$v_i$** tal que **$v_i \in W$, $v_k \in T$ e $(v_k, v_i) \in A$**

-Atribua a cada **$v_i$** um rótulo temporário igual a **dist($v_p$,$v_i$)**

-Se existe mais de uma distância para **$v_i$** então

- rótulo temporário de **$v_i$** = $\min(PL(v_k)+l_{ki})$, para todo **$v_k \in T$**

-Seja **v** o **$v_i$** com menor rótulo:

-Faça **v** vértice permanente transferindo-o de W para T

-Armazene em PL o rótulo de v ($PL=PL+\{TL(v)\}$)

-Armazene em P o vértice antecessor de v ($P=P+\{v_k:(v_k,v) \in A\}$)

-$TL=\{\infty$ qq $w_i$ pertencente $W\}$ (lembre-se $v \notin W$)

-Fim do enquanto

-Se $v_q \in T$ então

-A distância do menor caminho de **$v_p$** a **$v_q$** é dada por PL(**$v_q$**)

-Para encontrar o menor caminho propriamente dito,

basta encontrar **$v_q$** em **T** (ele é o último).

A partir dele, encontre o vértice correspondente em **P** (chame-o **$v_m$**). Ache **$v_m$** em **T**. Prossiga achando correspondentes aos $v_m$ em **P** e em **T** até chegar a **$v_p$**.

-Senão

-Não existe um caminho entre $v_p$ e $v_q$

-Fim

- L: matriz de distâncias
- V: conjunto de vértices do dígrafo
- T: vetor com os vértices permanentes
- PL: vetor com os labels permanentes
- W: vetor com os vértices ainda não-permanentes
- TL: vetor com rótulos temporários
- P: vetor com vértices ‘antecedentes’
- $v_p$ vértice inicial
- $v_q$ vértice final
- $l_{ij}$ distância entre os vértices i e j

# Floyd-Warshall

SE $i \neq j$ E $(i,j) \in A$ então $B_0[i,j]=C[i,j]$

SE $(i,j) \notin A$ então $B_0[i,j]=\infty$

SE $i=j$ então $B_0[i,j]=0$

Para $k=1$ até $N$ faça

$B_k[i,j]=\min(B_{k-1}[i,j], B_{k-i}[i,k]+B_{k-1}[k,j])$

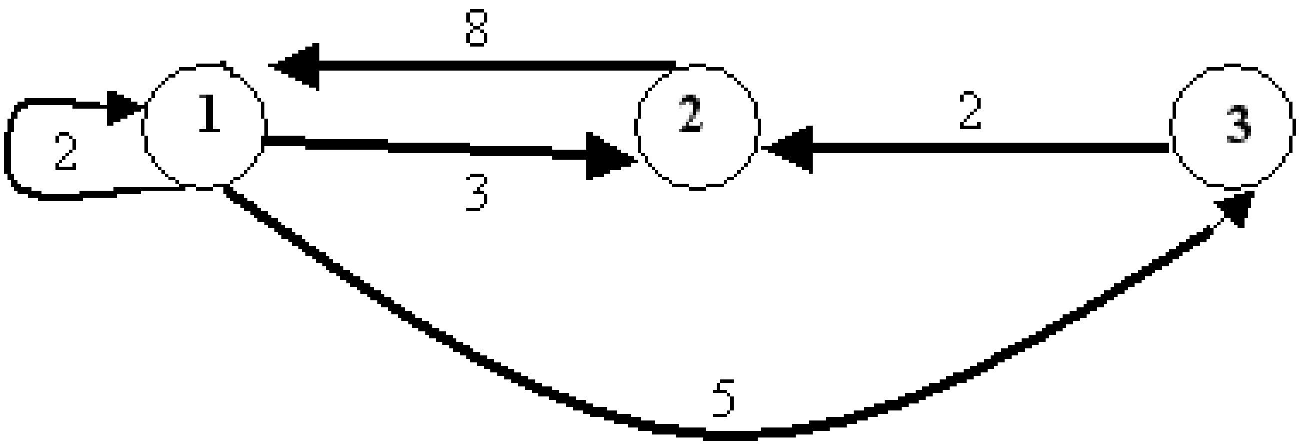
$B_n$ contém a distância dos caminhos mínimos de todos os pares de vértices

C[i,j]: custo para ir de i a j

A: conjunto de arestas do grafo

N: número de vértices do grafo

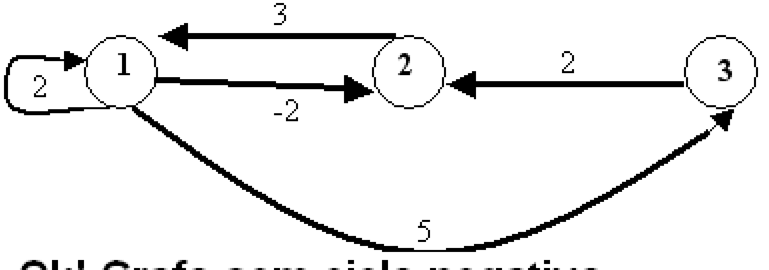
# Exemplo de Floyd-Warshall



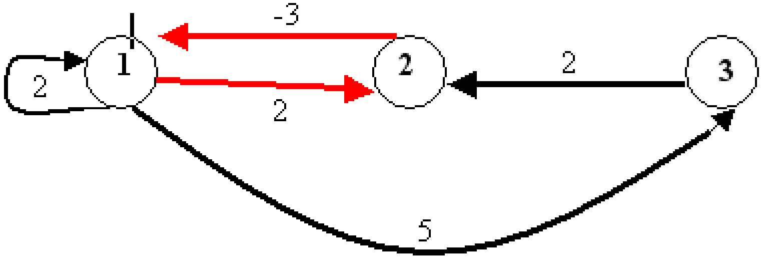

$$A_0 = \begin{bmatrix} 0 & 8 & 5 \\ 3 & 0 & \infty \\ \infty & 2 & 0 \end{bmatrix} \qquad A_1 = \begin{bmatrix} 0 & 8 & 5 \\ 3 & 0 & 8 \\ \infty & 2 & 0 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 0 & 8 & 5 \\ 3 & 0 & 8 \\ 5 & 2 & 0 \end{bmatrix} \qquad A_3 = \begin{bmatrix} 0 & 7 & 5 \\ 3 & 0 & 8 \\ 5 & 2 & 0 \end{bmatrix}$$

# Floyd-Warshall

- O algoritmo de Floyd-Warshall determina as distâncias dos menores caminhos entre todos os pares de vértices de um grafo
- Trabalha com arestas com pesos negativos
- Mas não funciona quando existem ciclos negativos no grafo




Ok! Grafo sem ciclo negativo




Nada feito. Grafo com ciclo negativo (arestas vermelhas)