

## 5. Caminhos Mínimos

↳ Grafos ponderados  $G = (V, A, w)$

↳ custo: distância, tempo, ... (mínimo)

↳ ~~Formar~~ caminhos de menor custo  
↳ encontrar

↳ Um caminho  $p = \langle v_1, v_2, \dots, v_k \rangle$

↳ custo mínimo (teórico) :  $\delta(u, v)$  é o custo mínimo teórico

• P/ um caminho impossível  $\delta(u, v) = \infty$

→ Problemas de caminho mínimo:

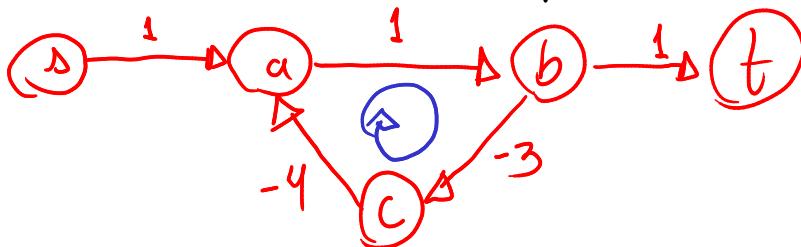
- Fonte única:  $G = (V, A, w) + s \in V$  (origem) → Caminhos mínimos a partir de  $s$  p/ todos
- Único destino:  $G + t \in V$  (destino) → Cam. mín. p/  $t$ .
- origem e destino definidos:  $G + s + t$  →  $p = \langle s, \dots, t \rangle$  é mínimo
- para todos os pares (origem, destino):  $G$  → todos os cam. mín. p/ cada  $s \in V + t \in V$ , encontra o cam. mín. de  $s$  para  $t$ .

↳ Floyd-Warshall

↳ Bellman-Ford / Dijkstra

→ Pesos negativos:

- alguns algoritmos operam sobre um grafo com arestas ou arcos com custo negativo. Nenhum dos que visitaremos opera sobre ciclo de custo negativo.



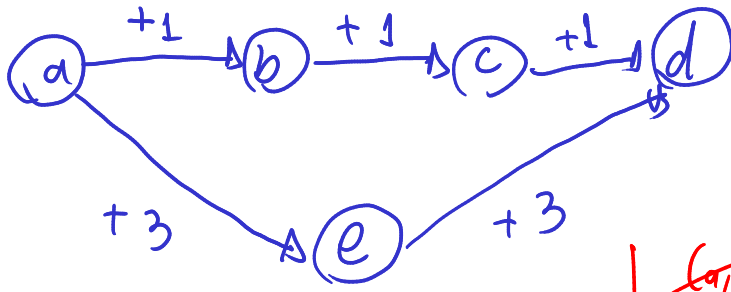
Custo:  $-\infty$

→ Pequenos algoritmos: inicialização e relaxamento.

⋮

→ Propriedade 5.1.6:

$\Delta = a$



$\Delta, t$   
 $\downarrow$   
 $\{a, b, c, d\}$

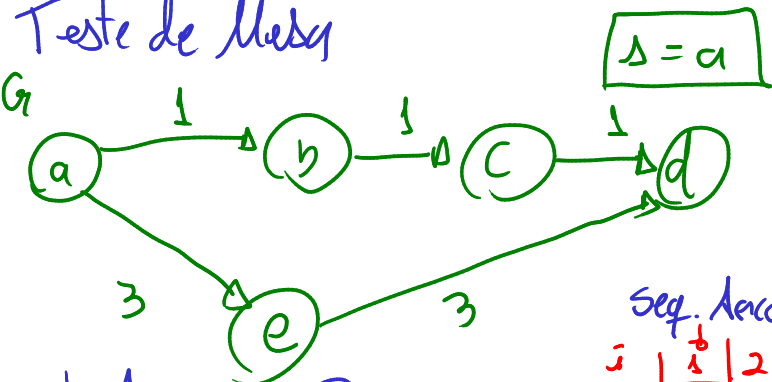
$D_a = 0$   
 $D_b = \cancel{\infty} 1$   
 $D_c = \cancel{\infty} 2$   
 $D_d = \cancel{\infty} 3$   
 $D_e = \infty$

~~(a,b)~~  
~~(b,c)~~  
~~(c,d)~~

## 5.2 Bellman-Ford

- Trata problema de fonte única;
- ~~Admite~~ arestas/arcs de custo negativo (ciclo negativo);
- Detecta se há ciclo negativo.

Teste de Mesas



	A	D
a	a	<del>∞</del> 0
b	<del>a</del>	<del>∞</del> 1
c	<del>b</del>	<del>∞</del> 2
d	<del>c</del>	<del>∞</del> 3
e	<del>a</del>	<del>∞</del> 3

$a \rightsquigarrow a / 0 / (a)$   
 $a \rightsquigarrow b / 1 / (a \rightarrow b)$

Seq. Arcos:

i	1	2	3	4	5
(e,d)	x	x	x	x	x
(c,d)	x	x	x	x	x
(b,c)	x	x	x	x	x
(a,e)	x	x	x	x	x
(a,b)	x	x	x	x	x

$a \rightsquigarrow c / 2 / (a \rightarrow b \rightarrow c)$   
 $a \rightsquigarrow d / 3 / (a \rightarrow b \rightarrow c \rightarrow d)$

}  $a \rightsquigarrow e / 3 / (a \rightarrow e)$

**Algoritmo 10:** Algoritmo de Bellman-Ford.  
**Input** : um grafo  $G = (V, E, w)$ , um vértice de origem  $s \in V$

```

// inicialização
1  $D_v \leftarrow \infty \forall v \in V$ 
2  $A_v \leftarrow \text{null} \forall v \in V$ 
3  $D_s \leftarrow 0$ 
4 for  $i \leftarrow 1$  to  $|V| - 1$  do
5   foreach  $(u, v) \in E$  do
6     // relaxamento
7     if  $D_v > D_u + w((u, v))$  then
8        $D_v \leftarrow D_u + w((u, v))$ 
9        $A_v \leftarrow u$ 
9   foreach  $(u, v) \in E$  do
10    if  $D_v > D_u + w((u, v))$  then
11      return (false, null, null)
12 return (true, D, A)
```