Aula 23

Circuito/Trajeto Euleriano

(Problema dos pontes de Konigsberg)

G não é simples:

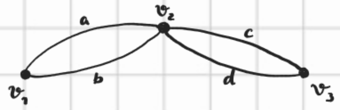
Não é possível!

D1 → Circuito Euleriano (circuito de Euler): trajeto fechado (não repete arestas) contento todas as arestas do grafo

T1 (Euler) → O grafo finito e conexo admite um circuito Euleriano se e só se todos os vértices de G têm grau par.

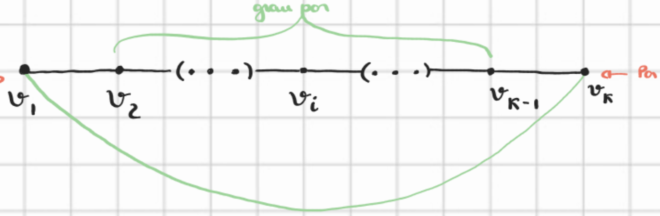
Demonstração: G circuito Euleriano ⟺ todos os vértices de G têm grau par

Multigrafo →



$C = \{ v_1, a, v_2, c, v_3, d, v_2, b, v_1)$

↳ todos os vértices têm grau par ⟺ G admite um circuito de Euler



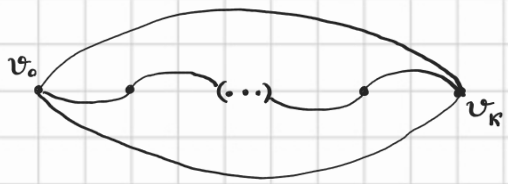
ex: ↳

É possível ↳

D2 → Trajeto Euleriano: trajeto (não repete arestas) contendo todas as arestas do grafo

T2 → O grafo finito e conexo admite um Trajeto Euleriano se e só se o número de vértices de grau ímpar é 0 ou 2

Demonstração: G trajeto Euleriano ⟺ o número de vértices de grau ímpar é 0 ou 2



→ v_0 e v_K são os únicos vértices de grau ímpar

trajeto Euleriano: $(v_0, \dots, v_K)$ ou $(v_K, \dots, v_0)$

Grafo bipartido $G = (V, E, \psi)$

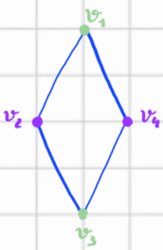
• Se $V = X \cup Y$, com $X \cap Y = \emptyset$

$\Rightarrow \underbrace{G[X] \text{ e } G[Y]}$ são grafos nulos!

não existem arestas entre vértices que estão em X ou entre vértices que estão em Y } bipartido $\{X, Y\}$ →

$\boxed{\begin{array}{l} G = (X, Y, E, \psi) \\ \text{ou} \\ G = (X, Y, E) \rightarrow \text{grafo simples} \end{array}}$

ex:

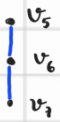


$X = \{v_1, v_3\}$
$Y = \{v_2, v_4\}$

Teorema: G com pelo menos 2 vértices é bipartido <u>se e só se</u> G não tem circuitos de comprimento ímpar

Demonstração: • G é bipartido $\Rightarrow$ G não tem circuitos de comp. ímpar

ex: $H = \{v_1, v_2, v_3, v_4\}$



$H[v_5, v_6, v_7]$ bipartido

$H = (X, Y, E)$ $X = \{v_1, v_3, v_5, v_7\}$
$Y = \{v_2, v_4, v_6\}$

X :

Y :

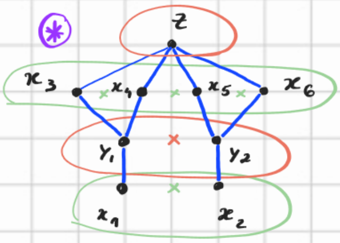
para voltar tem sem de ter sempre comprimento par

→ $(\Leftarrow)$

• G não tem circuitos com comp. ímpar $\Longrightarrow$ G é bipartido

Supondo que G é anexo a qualquer par de vértices, é unido por um caminho. Senão, o procedimento pode aplicar-se a qualquer componente conexa
Um grafo é bipartido <u>se e só se</u> cada um dos seus componentes conexos é um subgrafo bipartido

ex: Seja z um vértice de G e seja $X = \{x \in V_G : dist_G(z, x) \text{ é ímpar}\}$



$\boxed{\begin{array}{c} \underline{\text{Par}} \\ I + I + 1 = I \\ P + P + 1 = I \end{array}}$

Facto 1: Não existem arestas que ligam vértices de x (caso contrário teríamos circuitos de comprimento ímpar)

Facto 2: Todos os vértices de $V_G \setminus X$ estão a uma distância par de z, pelo que não existem arestos que ligem vértices em $V_G \setminus X$ (caso contrário teríamos circuitos de comprimento ímpar)

Logo: G é bipartido onde $(X, V_G \setminus X)$ é a bipartição dos vértices

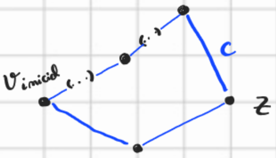
Por exemplo:

$Y = \{Y_1, Y_2, z\}$

$X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$

Algoritmo de Dijkstra → IMPORTANTE!

- Serve para o cálculo de menos custo (de custo mínimo) entre dois vértices



Só custos positivos!

- $\text{marca}(u) > \text{marca}(z) + \underbrace{W_{zu}}_{c \in \mathbb{R}^+}$
- $\text{antecessor}(u) = z$

Os pares têm de ter sempre → $\big(\text{marca}(u), \text{antecessor}(u)\big)$

O desenvolvimento

- Inicializar as variáveis:
  - Para cada $v \in V$: **marca**$(v) = \infty$, **ant**$(v) = \varnothing$.
  - **marca**$(\texttt{start}) = 0$.
  - **temp** $= V \setminus \{\texttt{start}\}$ e menor $= \texttt{start}$.
- Repetir:
  - $c_{\text{aux}} = \infty$.
  - Para todo o v em **temp**:
    - Se **marca**$(v) >$ **marca**$(\texttt{menor}) + W(\texttt{menor}, v)$, então

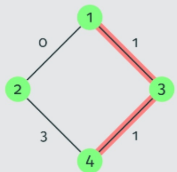
      marca$(v) =$ **marca**$(\texttt{menor}) + W(\texttt{menor}, v)$,
      ant$(v) = \texttt{menor}$.
    - Se **marca**$(v) < c_{\text{aux}}$ então $c_{\text{aux}} =$ **marca**(v) e $v_{\text{aux}} = v$ (lembrar do «menor custo»).
  - **temp** $=$ **temp** $\setminus \{v_{\text{aux}}\}$ e menor $= v_{\text{aux}}$.

  Até menor $=$ o vértice terminal.

Exemplo

não existe antecessor

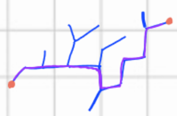
1	2	3	4	menor	**temp**
$(0,-)$	$(\infty,-)$	$(\infty,-)$	$(\infty,-)$	1	$\{2,3,4\}$
	$(0,1)$	$(1,1)$	$(\infty,-)$	2	$\{3,4\}$
		$(1,1)$	$(3,2)$	3	$\{4\}$
			$(2,3)$	4	$\varnothing$



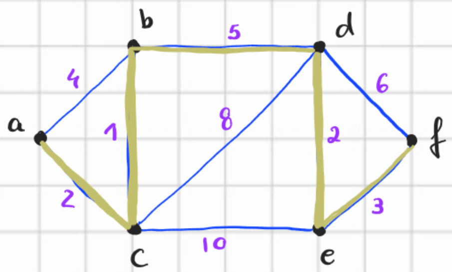
- vértice inicial: 1.
- vértice terminal: 4.
- Notação: (custo, vértice anterior).

- **marca**(v) = «custo» do caminho de menor «custo» entre start e v (até o momento).
- **ant**(v) = antecessor de $v \neq$ start no caminho de menor «custo» entre start e v (até o momento).

Implementação em java: (em árvore)



ex:



Matriz de custos: →

$$
M = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array}
\begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\
\left[\begin{array}{cccccc}
0 & 4 & 2 & \infty & \infty & \infty \\
4 & 0 & 1 & 5 & \infty & \infty \\
2 & 1 & 0 & 8 & 10 & \infty \\
\infty & 5 & 8 & 0 & 2 & 6 \\
\infty & \infty & 10 & 2 & 0 & 3 \\
\infty & \infty & \infty & 6 & 3 & 0
\end{array}\right]
\end{array}
$$

não é possível!

- Caminho de menor custo entre os vértices a e f

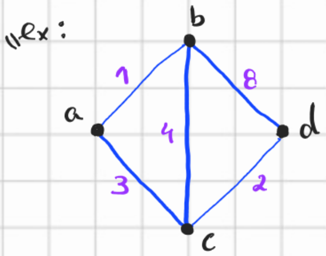
a	b	c	d	e	f	menor	temp
(0,-)	(∞,-)	(∞,-)	(∞,-)	(∞,-)	(∞,-)	a	{b,c,d,e,f}
	(4,a)	(2,a)	(∞,-)	(∞,-)	(∞,-)	c	{c,d,e,f}
	(3,c)		(10,c)	(12,c)	(∞,-)	b	{d,e,f}
			(8,d)	12<∞ (12,c)	(∞,-)	d	{e,f}
				10<12 (10,d)	(14,d)	e	{f}
					(13,e)	f	∅

Acumula-se!

$C_m = (a, c, b, d, e, f)$ de custo $w(C_m) = 13$

<u>Nota</u>: Se houver dois <u>menores</u> iguais escolhemos 1!

"ex:



$C_m = (a, c, d)$

T.P.C!