

# Edsger Wybe Dijkstra (1930 – 2002)

1

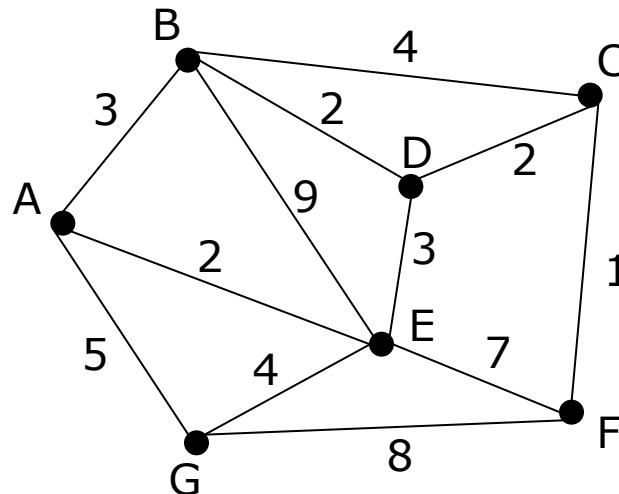
- Algoritmos, grafos, linguagens de programação, compiladores, sistemas operacionais e distribuídos, programação concorrente...
- *A pronúncia aproximada em português para Edsger Dijkstra é étsrrar déikstra.*

[https://pt.wikipedia.org/wiki/Edsger\\_Dijkstra](https://pt.wikipedia.org/wiki/Edsger_Dijkstra)



# Caminho Mais Curto

- Considera-se o **peso** dos arcos como **distâncias** entre os vértices.
  - O caminho mais curto **não** é o menor número de arcos entre os dois vértices!
- **Como encontrar o caminho mais curto entre dois vértices quaisquer?**
  - Obs.: A árvore geradora mínima é a resposta para a pergunta: qual o caminho mais curto entre todos os vértices?
- **Ex.:** Qual o caminho mais curto entre os vértices **A** e **F**?



# Algoritmo de Dijkstra

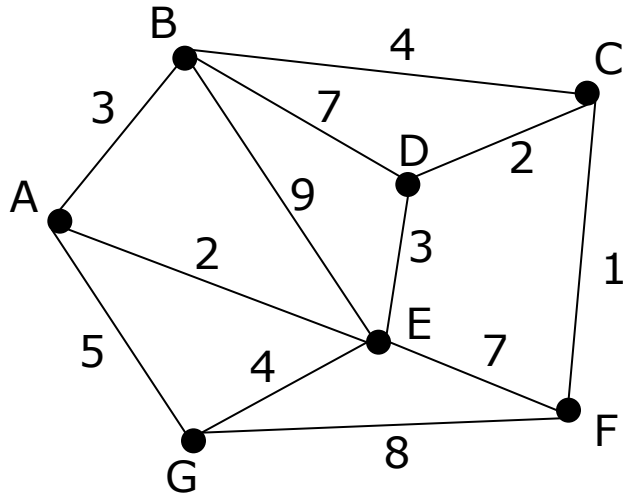
- **Considerações:**

- O algoritmo trabalha “rotulando” os vértices do grafo.
- O rótulo de um vértice indica a distância entre aquele vértice e o vértice inicial.
- São utilizados rótulos *permanentes* e rótulos *temporários*.
- Rótulos temporários podem mudar de valor durante a execução do algoritmo, pois ele busca encontrar o caminho mais curto.
- O rótulo *permanente* de um vértice indica a menor distância entre aquele vértice e o vértice inicial.
- O algoritmo termina quando o vértice final recebe um rótulo *permanente*.

- **Estrutura de dados:** matriz de adjacências,  $m$ , tal que

$$m_{ij} = \begin{cases} \text{peso do arco entre } i \text{ e } j, \text{ se existir} \\ 0, \text{ se } i = j \\ \infty, \text{ se } \mathbf{n\tilde{a}o} \text{ existe arco entre } i \text{ e } j \end{cases}$$

# Algoritmo de Dijkstra



Matriz de Adjacências:

	A	B	C	D	E	F	G
A	0	3	$\infty$	$\infty$	2	$\infty$	5
B	3	0	4	7	9	$\infty$	$\infty$
C	$\infty$	4	0	2	$\infty$	1	$\infty$
D	$\infty$	7	2	0	3	$\infty$	$\infty$
E	2	9	$\infty$	3	0	7	4
F	$\infty$	$\infty$	1	$\infty$	7	0	8
G	5	$\infty$	$\infty$	$\infty$	4	8	0

- **Algoritmo é dividido em três blocos:**

- 1) **Inicialização:** inicialização dos rótulos dos vértices
- 2) **Rotulação:** atribuição de rótulos permanentes a cada vértice (sua menor distância ao vértice inicial)
- 3) **Montagem do caminho** mais curto entre os vértices inicial e final

# Algoritmo de Dijkstra

- **Inicialização:**

- Rotula o vértice inicial com o rótulo permanente 0 e todos os outros com rótulo temporário  $\infty$ .

- **Rotulação:**

**enquanto** vértice final não tem rótulo permanente **faça**

**para** todo vértice  $v$  *sem* rótulo permanente **faça**

rotule o vértice  $v$  com rótulo temporário dado por

**min**(rótulo de  $v$ , (rótulo de  $i$  +  $m_{iv}$ ))

onde  $i$  é o último vértice a receber rótulo permanente e

$m_{iv}$  é o peso do arco entre  $i$  e  $v$

**fim para**

encontre o menor valor entre os rótulos *temporários* e o faça permanente para aquele vértice

**fim enquanto**

# Algoritmo de Dijkstra

---

- **Montagem do caminho:**

- Retorne do vértice final (primeiro “vértice corrente”) para trás:
- **Enquanto** o vértice corrente for diferente do vértice inicial **faça**
  - Escolha o vértice cuja diferença entre o rótulo do vértice corrente e o dele seja exatamente igual ao peso do arco entre eles.
  - (Se existe mais de um vértice satisfazendo a condição acima, indica que há mais de um caminho mais curto)
  - Faça o vértice escolhido ser o novo vértice corrente.

# Algoritmo de Dijkstra: exemplo 1

**enquanto** vértice final não tem rótulo permanente **faça**

**para** todo vértice  $v$  *sem* rótulo permanente **faça**

rotule o vértice  $v$  com rótulo temporário dado por

**min**(rótulo de  $v$ , (rótulo de  $i$  +  $m_{iv}$ ))

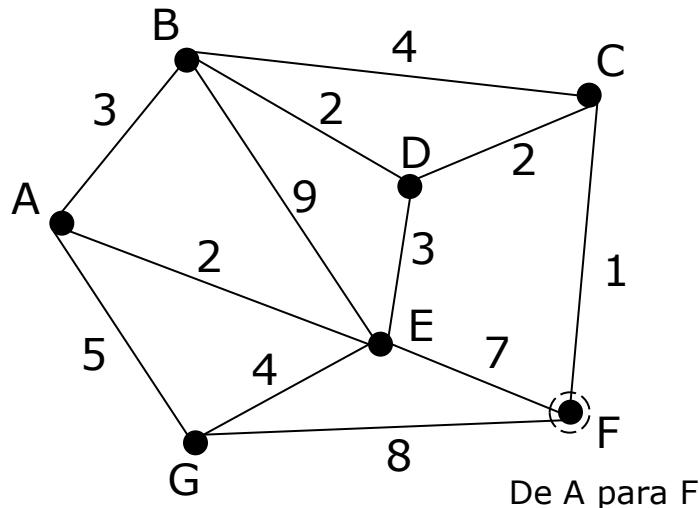
onde  $i$  é o último vértice a receber rótulo permanente e

$m_{iv}$  é o peso do arco entre  $i$  e  $v$

**fim para**

encontre o menor valor entre os rótulos *temporários* e o faça permanente para aquele vértice

**fim enquanto**



i	v	permanente	temporário
	<b>A</b>	<u><b>0</b></u>	<b>A</b>
	<b>B</b>		<b>B</b>
	<b>C</b>		<b>C</b>
	<b>D</b>		<b>D</b>
	<b>E</b>		<b>E</b>
	<b>F</b>		<b>F</b>
	<b>G</b>		<b>G</b>

# Algoritmo de Dijkstra: exemplo 1

**enquanto** vértice final não tem rótulo permanente **faça**

**para** todo vértice  $v$  *sem* rótulo permanente **faça**

        rotule o vértice  $v$  com rótulo temporário dado por

**min**(rótulo de  $v$ , (rótulo de  $i$  +  $m_{iv}$ ))

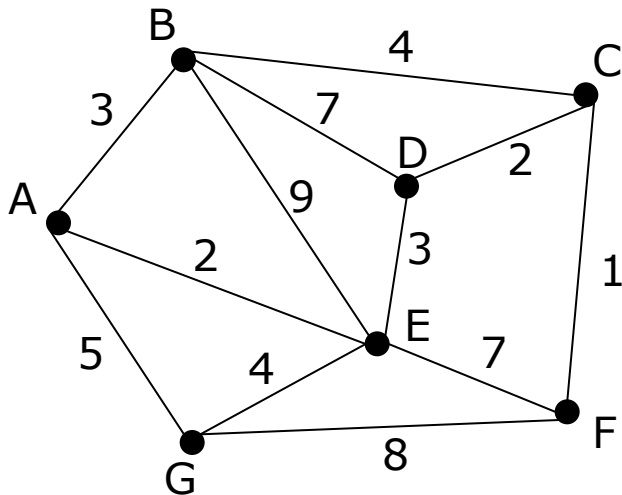
        onde  $i$  é o último vértice a receber rótulo permanente e

$m_{iv}$  é o peso do arco entre  $i$  e  $v$

**fim para**

    encontre o menor valor entre os rótulos *temporários* e o faça permanente para aquele vértice

**fim enquanto**



<b>A</b>	<u><b>0</b></u>					
<b>B</b>	$\infty$	3	<u><b>3</b></u>			
<b>C</b>	$\infty$	$\infty$	$\infty$	7	7	<u><b>7</b></u>
<b>D</b>	$\infty$	$\infty$	5	<u><b>5</b></u>		
<b>E</b>	$\infty$	<u><b>2</b></u>				
<b>F</b>	$\infty$	$\infty$	9	9	9	9 <u><b>8</b></u>
<b>G</b>	$\infty$	5	5	5	<u><b>5</b></u>	



# Algoritmo de Dijkstra: exemplo 2

**enquanto** vértice final não tem rótulo permanente **faça**

**para** todo vértice  $v$  *sem* rótulo permanente **faça**

        rotule o vértice  $v$  com rótulo temporário dado por

**min**(rótulo de  $v$ , (rótulo de  $i$  +  $m_{iv}$ ))

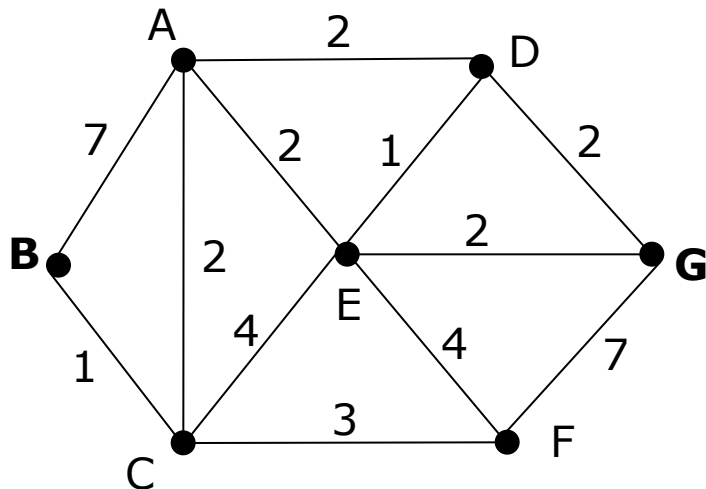
        onde  $i$  é o último vértice a receber rótulo permanente e

$m_{iv}$  é o peso do arco entre  $i$  e  $v$

**fim para**

        encontre o menor valor entre os rótulos *temporários* e o faça permanente para aquele vértice

**fim enquanto**



**A**  $\infty$

**B** 0

**C**  $\infty$

**D**  $\infty$

**E**  $\infty$

**F**  $\infty$

**G**  $\infty$

# Algoritmo de Dijkstra: “origem” para “destino”

**enquanto** vértice final não tem rótulo permanente **faça**

**para** todo vértice  $v$  *sem* rótulo permanente **faça**

    rotule o vértice  $v$  com rótulo temporário dado por

$\min(\text{rótulo de } v, (\text{rótulo de } i + m_{iv}))$

    onde  $i$  é o último vértice a receber rótulo permanente e

$m_{iv}$  é o peso do arco entre  $i$  e  $v$

**fim para**

  encontre o menor valor entre os rótulos *temporários* e o faça permanente para aquele vértice

**fim enquanto**

