



# Algoritmo Dijkstra

Antônio Erick Freitas Ferreira

João Pedro Soares Matias

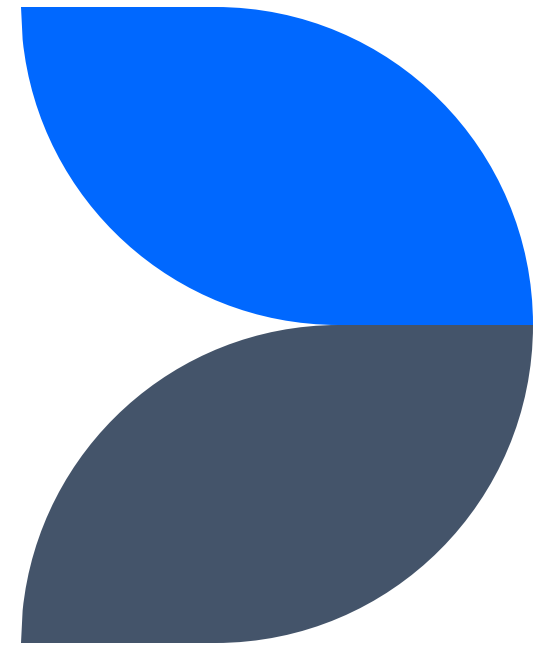


# Tópicos abordados:

- O que é o algoritmo de Dijkstra?
- Quais as aplicações desse algoritmo?
- Como o algoritmo funciona?
- Exemplo de funcionamento.
- Prova que o algoritmo Dijkstra encontra a solução ótima.
- Complexidade do algoritmo.
- Versão do algoritmo em C++.

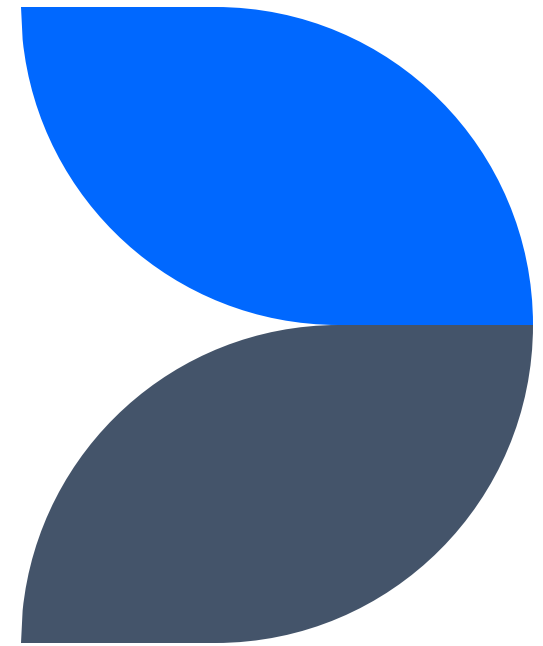


O que é o  
algoritmo de  
Dijkstra?



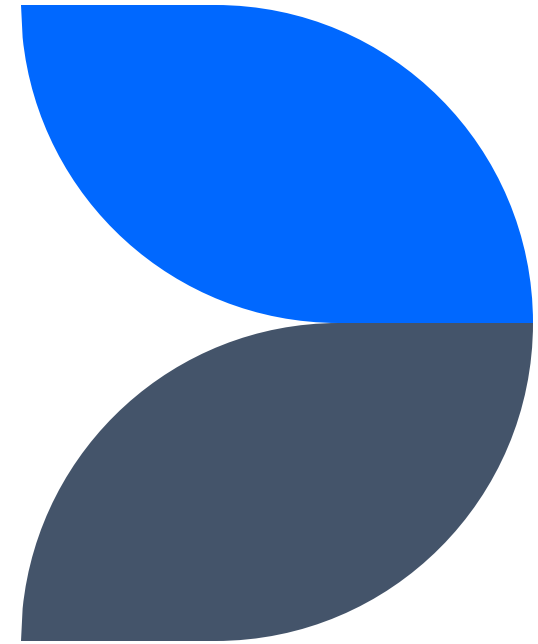
É um algoritmo utilizado para descobrir o caminho mínimo entre dois vértices em um grafo ponderado direcionado(ou não) que seja conexo e que não tenha ciclos negativos entre seus vértices.

Quais as  
aplicações desse  
algoritmo?



- Roteamento em redes de computadores
- Navegação em mapas
- Projeto de circuitos elétricos
- Análise de redes sociais

Como o algoritmo  
funciona?

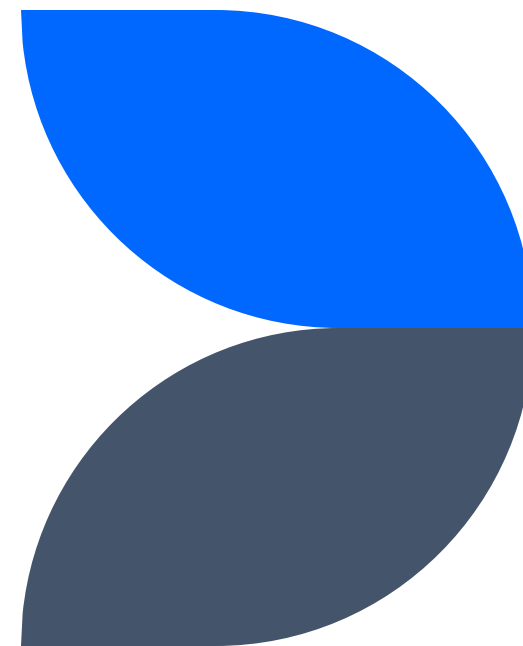


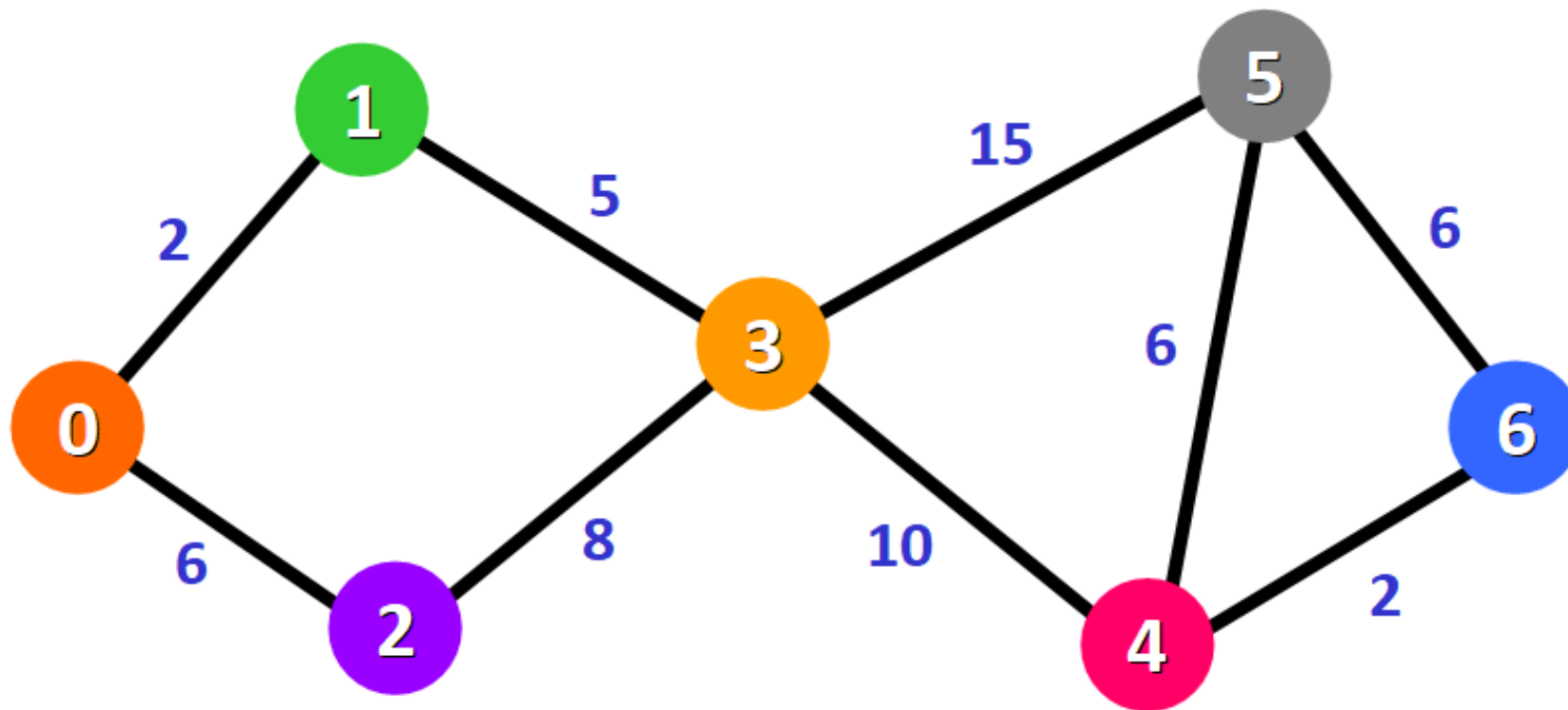
### **Dijkstra( $G, w, s$ )**

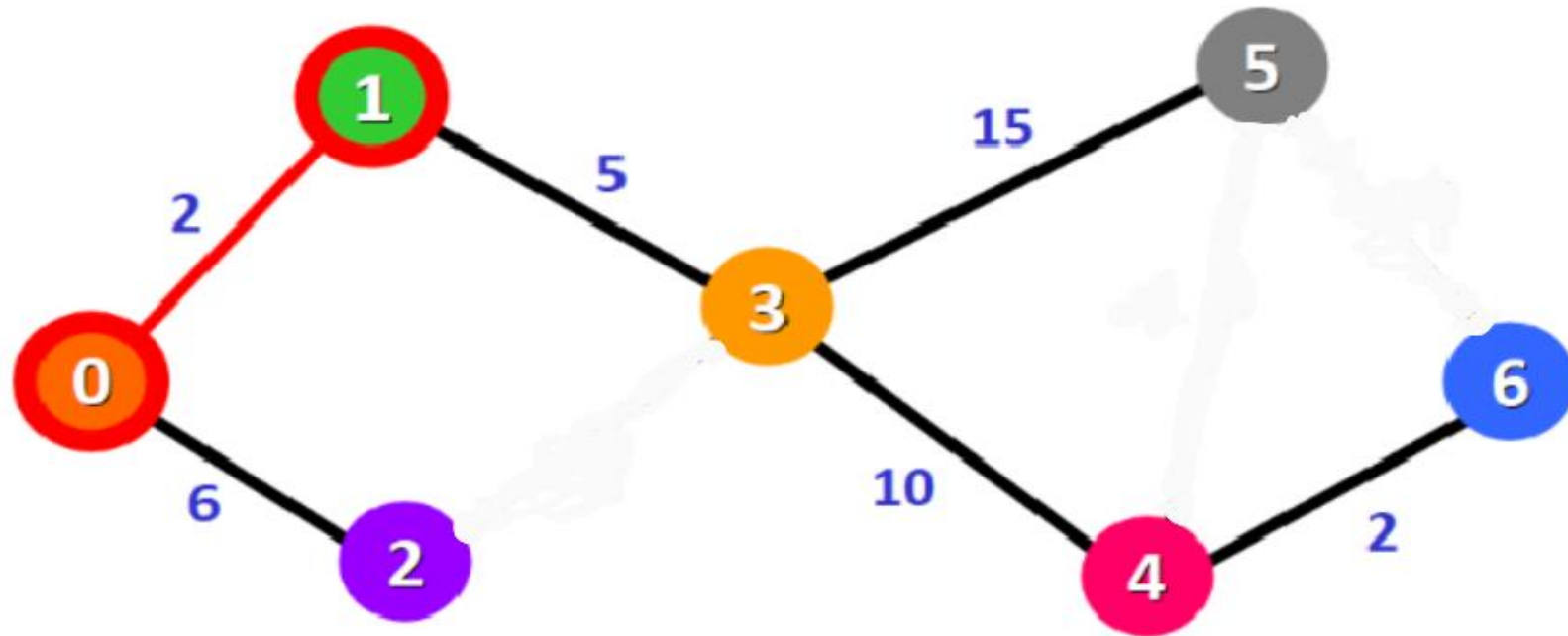
- 1) Initialize-Single-Source( $G, s$ )
- 2)  $S \leftarrow \emptyset$
- 3)  $Q = V[G]$
- 4) while  $Q \neq \emptyset$ 
  - 5) do  $u \leftarrow \text{Extract\_min}(Q)$
  - 6)  $S \leftarrow S \cup \{u\}$
  - 7) for each  $(u, v) \in \text{Adj}[u]$
  - 8) do Relax( $u, v, w$ )



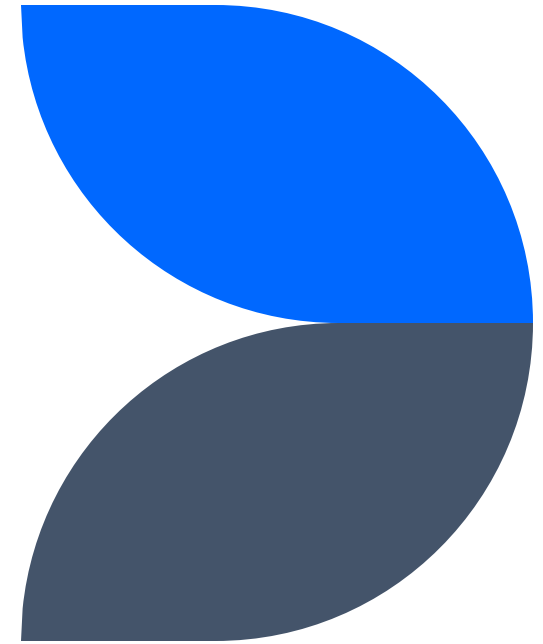
**Exemplo de  
funcionamento**







**Prova que o  
algoritmo  
encontra a  
solução ótima.**



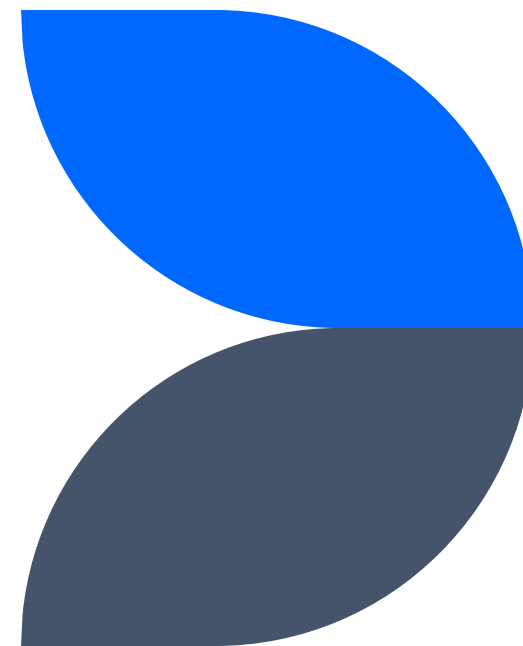
**Teorema: dado um vértice  $s \in G$  a  $\text{dist}(s,v)$  é a mínima.**

Caso base: Com  $i = 0$  o único vértice visitado é o vértice fonte  $s$ , onde  $s.\text{distancia} = 0$ .

Hipótese indutiva: Para todo vértice já visitado, o teorema é verdadeiro.

Passo indutivo: Na  $i$ -ésima iteração, o algoritmo irá escolher o vértice  $x$ .

**Complexidade do  
algoritmo.**



### Dijkstra( $G, w, s$ )

- 1) Initialize-Single-Source( $G, s$ )
- 2)  $S \leftarrow \emptyset$
- 3)  $Q = V[G]$
- 4) while  $Q \neq \emptyset$
- 5)     do  $u \leftarrow \text{Extract\_min}(Q)$
- 6)      $S \leftarrow S \cup \{u\}$
- 7)     for each  $(u, v) \in \text{Adj}[u]$
- 8)     do Relax( $u, v, w$ )

Na linha 1, a complexidade é  $O(|V|)$ .

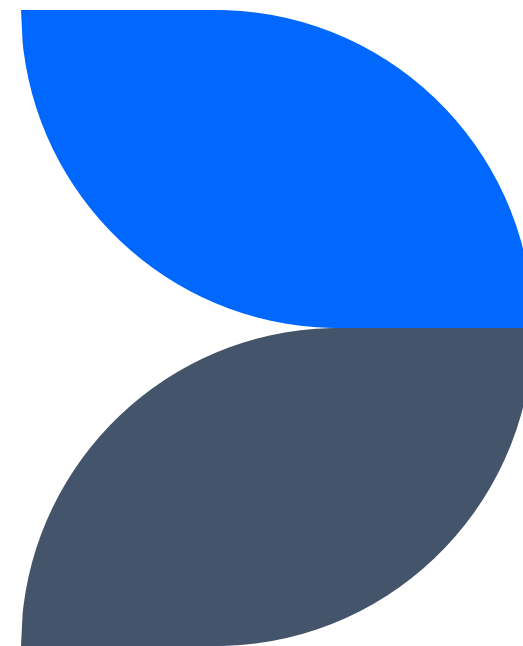
Na linha 3, a complexidade é  $O(|V|)$ .

No primeiro loop, a complexidade de remover um vértice da heap é  $O(\log |V|)$  somado com a complexidade do segundo loop interno que é  $O(|V| + |E|)$  + a complexidade da linha 8, que é  $O(1)$ . Temos que  $O(\log |V|) * O(|V| + |E|)$ .

Como  $|E| \gg |V|$  podemos considerar  $O(\log |V|) * O(|E|)$ .

Logo a complexidade final do algoritmo é  $O(|E| * \log |V|)$ .

Versão do  
algoritmo em  
C++.





“

Testes de programas podem ser uma maneira muito eficaz para demonstrar a presença de erros, mas é irremediavelmente insuficiente para mostrar a sua ausência.

”

Edsger Dijkstra.



# Obrigado

Antônio Erick

[erickdev1218@alu.ufc.br](mailto:erickdev1218@alu.ufc.br)

João Pedro Soares

[joao.pedrosm@alu.ufc.br](mailto:joao.pedrosm@alu.ufc.br)