

# ALGORITMOS EM GRAFOS

## CAMINHAMENTOS

### ALGORITMO DE DIJKSTRA

Prof. Alexei Machado

# Edsger Wybe Dijkstra (1930 – 2002)

2

- 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)

# Algoritmo de Dijkstra

3

- Baseado na busca em largura
- Encontra a menor distância entre dois vértices de um grafo ponderado
  - *encontra o menor caminho entre um vértice  $v_i$  e todos os demais vértices do grafo*

# Algoritmo de Dijkstra

4

- Definir um vértice de origem  $v_o$
- Utiliza um vetor de distâncias a partir de  $v_o$
- Utiliza um vetor de caminhos a partir de  $v_o$
- Utiliza um vetor de vértices não visitados do grafo

# Algoritmo de Dijkstra

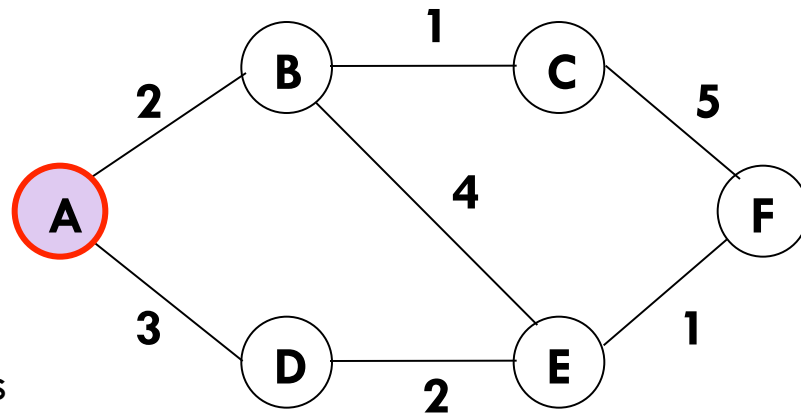
5

- Inicializações do algoritmo
  - $d[v_o] = 0$
  - Se existe  $a[v_o, v_i]$ ,  $d[v_i] = \text{peso}(v_o, v_i)$ 
    - Inserir  $v_o - v_i$  no vetor de caminhos
  - Se não existe  $a[v_o, v_i]$ ,  $d[v_i] = \text{max\_value}$

# Algoritmo de Dijkstra

G

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
    **faça**  
        (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$   
        (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)



Distâncias

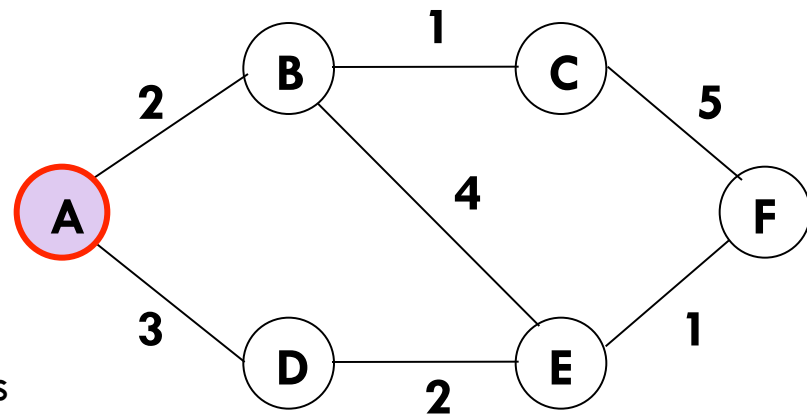
| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |

Caminhos

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | - | 3 | - | - | - |

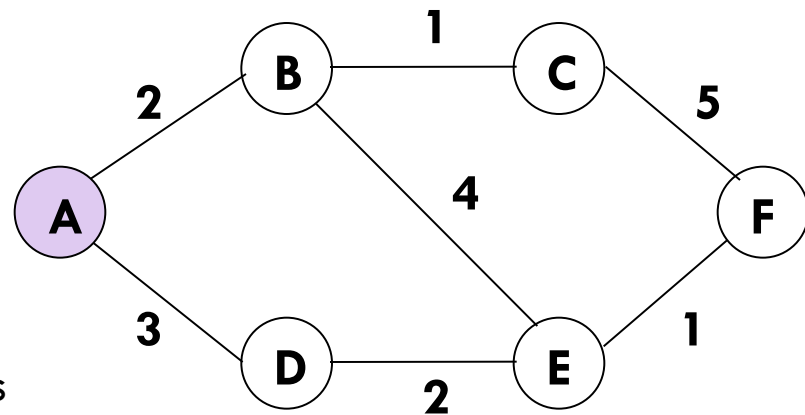
Caminhos

| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | - | 3 | - | - | - |

Caminhos

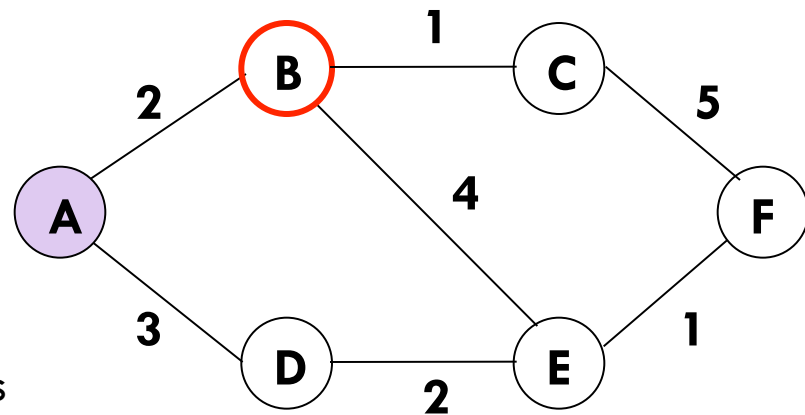
| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)



# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | - | 3 | - | - | - |

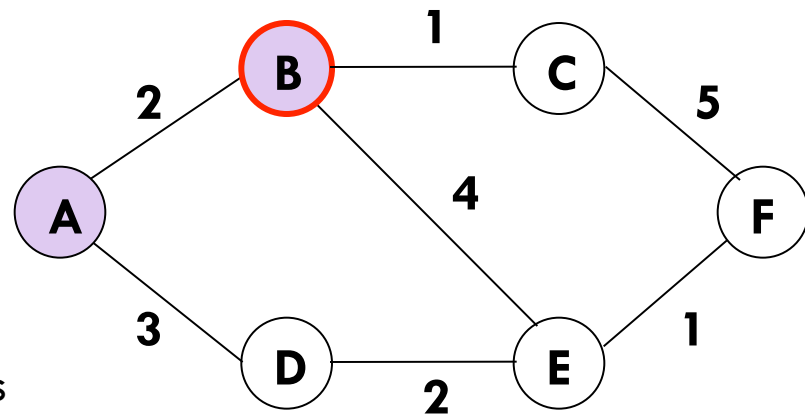
Caminhos

| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | - | 3 | - | - | - |

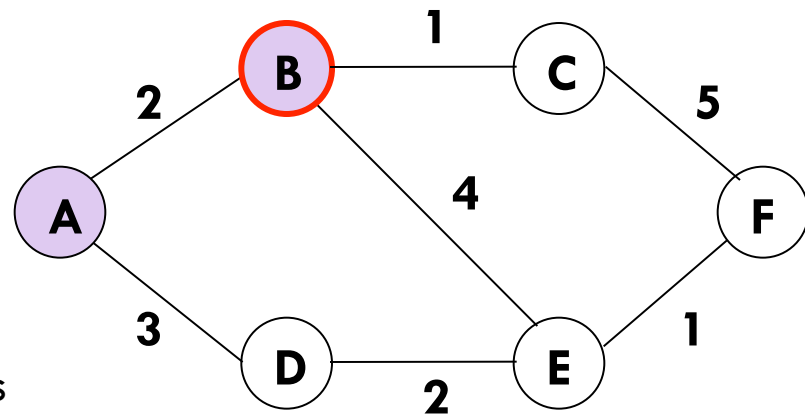
Caminhos

| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | - | 3 | - | - | - |

Caminhos

| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

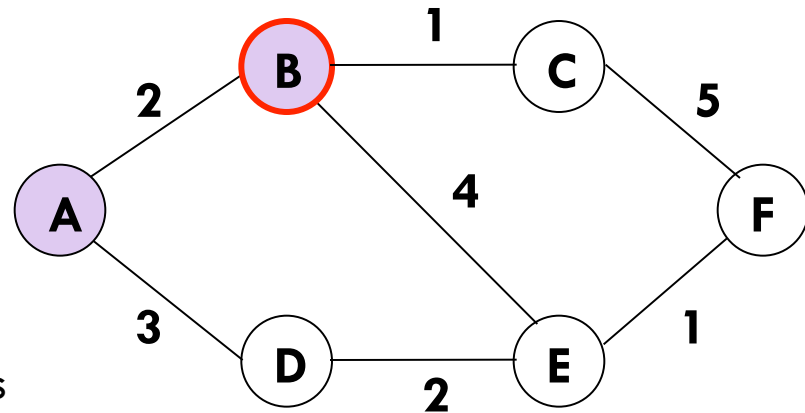
# Algoritmo de Dijkstra

G

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$  ○  

**faça**

  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 |   |   |   |   |   |   |

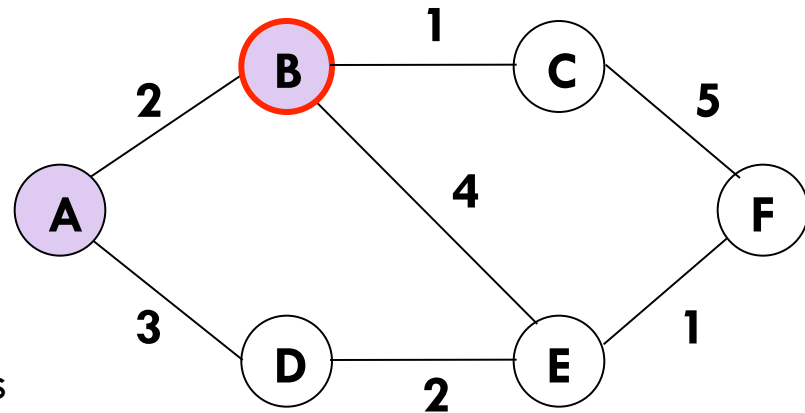
São os  
vértices C e E

| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

# Algoritmo de Dijkstra

G

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$  ○  
     **faça**  
     (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$   
     (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)



Distâncias

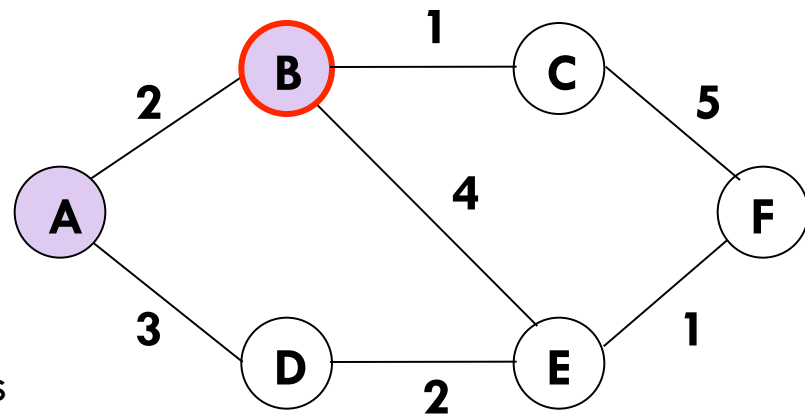
| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 |   |   |   |   |   |   |

| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

Começando por C

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | - | 3 | - | - | - |

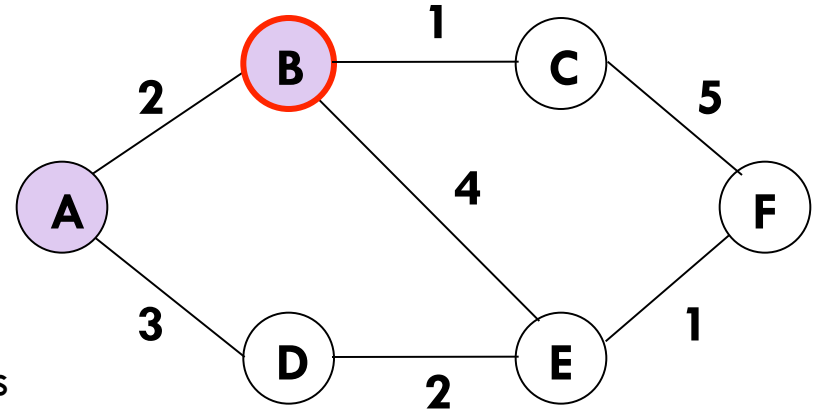
Caminhos

| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(A,B) + \text{aresta}(B,C) < d(A,C)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x,i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | - | 3 | - | - | - |

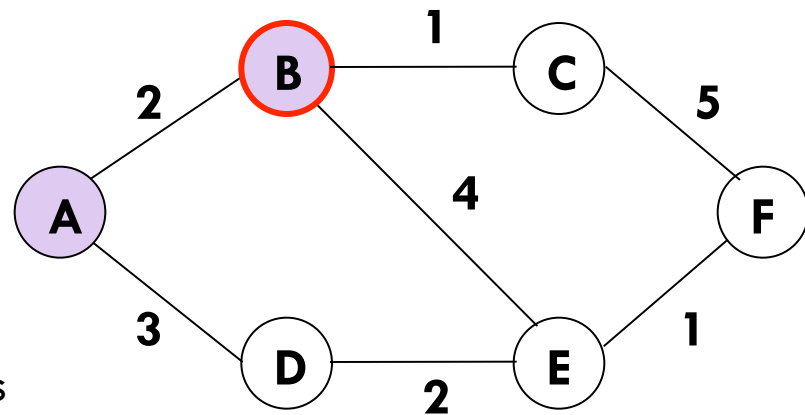
Caminhos

| A | B  | C | D  | E | F | G |
|---|----|---|----|---|---|---|
| A | AB |   | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(A,B) + \text{aresta}(B,C) < d(A,C)$   
**faça**
  - (i)  $d(A, C) = d(A, B) + \text{aresta}(B,C)$
  - (ii)  $c(C) = c(B) + C$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | - | - | - |

Caminhos

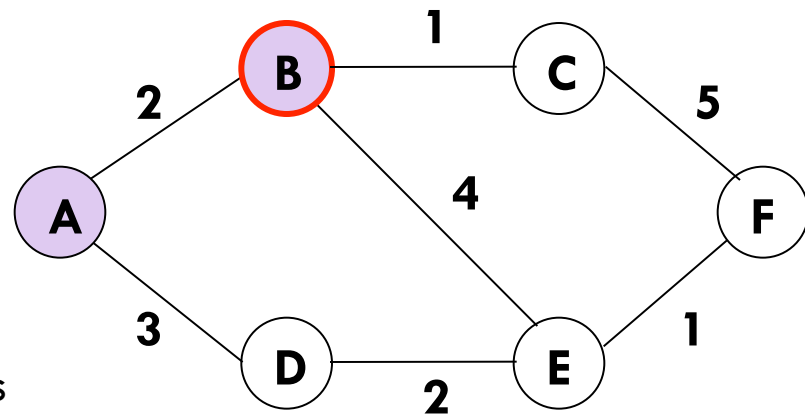
| A | B  | C   | D  | E | F | G |
|---|----|-----|----|---|---|---|
| A | AB | ABC | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(A,B) + \text{aresta}(B,C) < d(A,C)$   
**faça**
  - (i)  $d(A, C) = d(A, B) + \text{aresta}(B,C)$
  - (ii)  $c(C) = c(B) + C$
- (5) Se existirem vértices não visitados voltar para o passo (2)



# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 |   |   |   |   |   |   |

Agora, para E

| A | B  | C   | D  | E | F | G |
|---|----|-----|----|---|---|---|
| A | AB | ABC | AD |   |   |   |

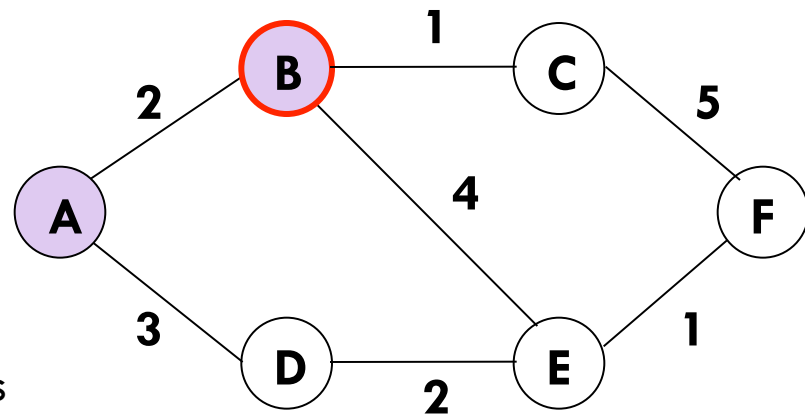
- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(A,B) + \text{aresta}(B,C) < d(A,C)$    

faça

  - (i)  $d(A, C) = d(A, B) + \text{aresta}(B,C)$
  - (ii)  $c(C) = c(B) + C$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | - | - | - |

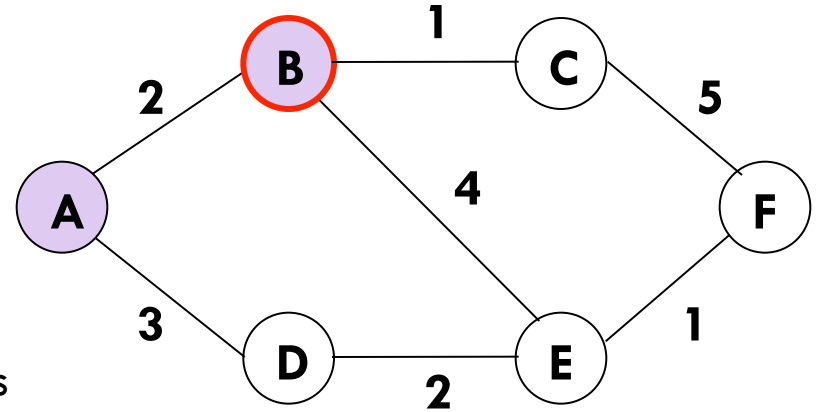
Caminhos

| A | B  | C   | D  | E | F | G |
|---|----|-----|----|---|---|---|
| A | AB | ABC | AD |   |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(A,B) + \text{aresta}(B,E) < d(A,E)$   
**faça**
  - (i)  $d(A, E) = d(A, B) + \text{aresta}(B,E)$
  - (ii)  $c(E) = c(B) + E$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 6 | - | - |

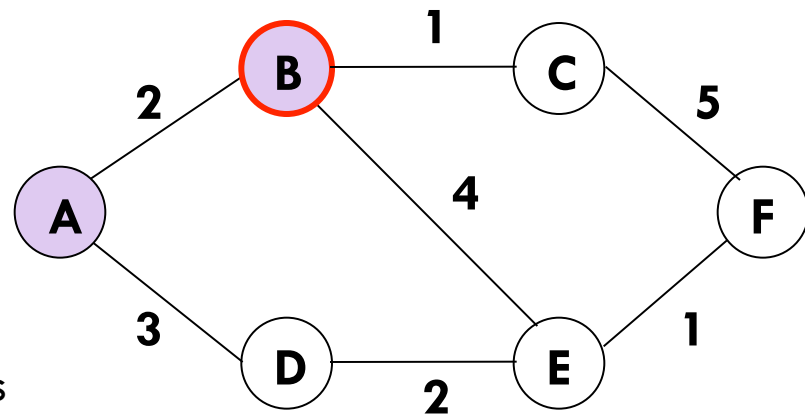
Caminhos

| A | B  | C   | D  | E   | F | G |
|---|----|-----|----|-----|---|---|
| A | AB | ABC | AD | ABE |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(A,B) + \text{aresta}(B,E) < d(A,E)$   
**faça**
  - (i)  $d(A, E) = d(A, B) + \text{aresta}(B,E)$
  - (ii)  $c(E) = c(B) + E$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 6 | - | - |

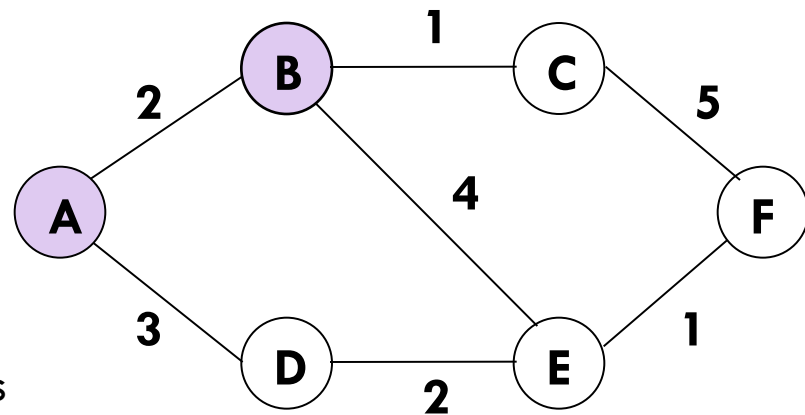
Caminhos

| A | B  | C   | D  | E   | F | G |
|---|----|-----|----|-----|---|---|
| A | AB | ABC | AD | ABE |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 6 | - | - |

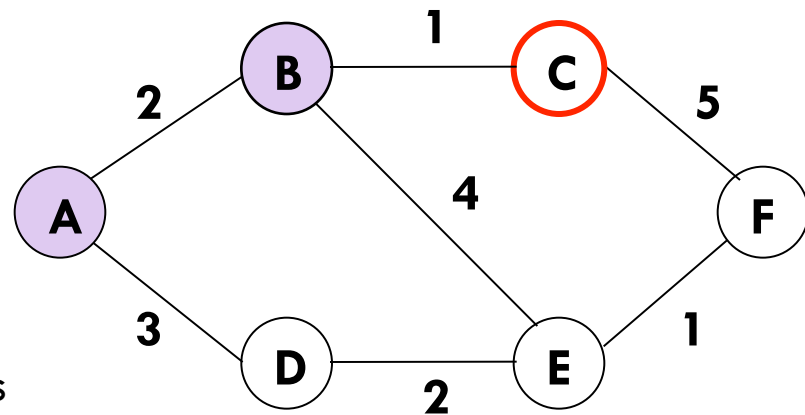
Caminhos

| A | B  | C   | D  | E   | F | G |
|---|----|-----|----|-----|---|---|
| A | AB | ABC | AD | ABE |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 6 | - | - |

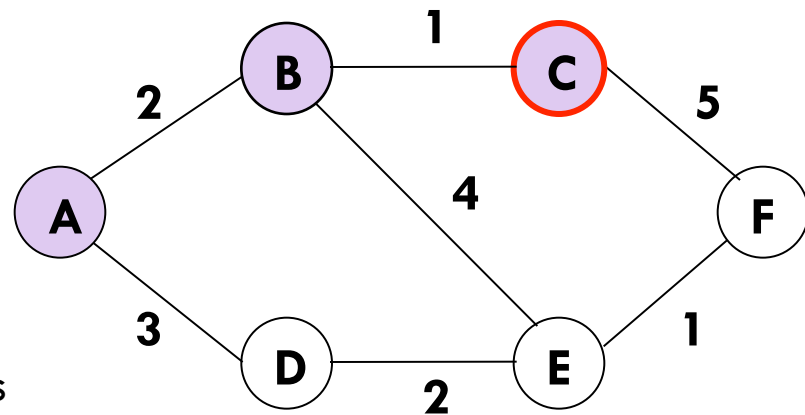
Caminhos

| A | B  | C   | D  | E   | F | G |
|---|----|-----|----|-----|---|---|
| A | AB | ABC | AD | ABE |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 6 | - | - |

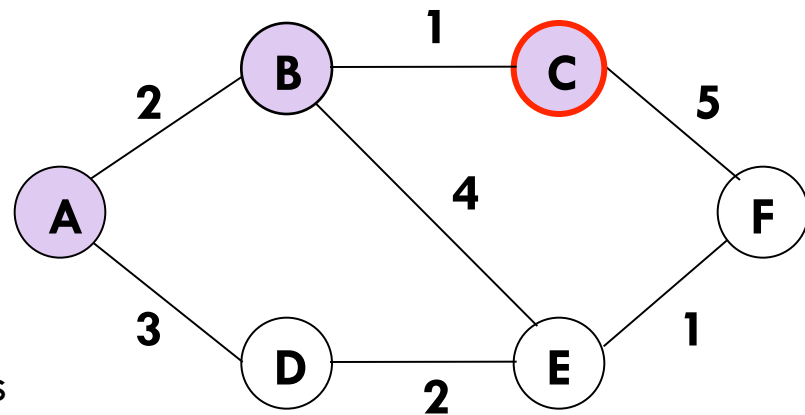
Caminhos

| A | B  | C   | D  | E   | F | G |
|---|----|-----|----|-----|---|---|
| A | AB | ABC | AD | ABE |   |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 6 | 8 | - |

Caminhos

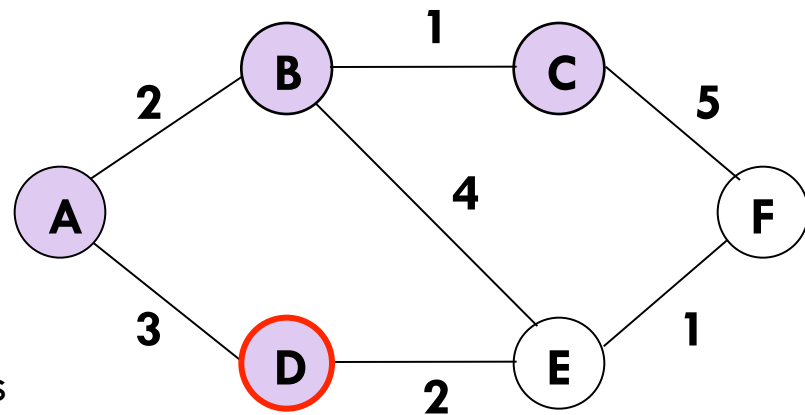
| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ABE | ABCF |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)



# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 6 | 8 | - |

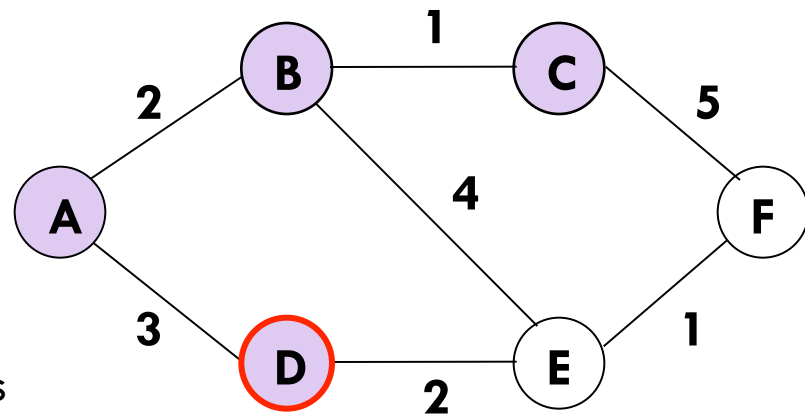
Caminhos

| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ABE | ABCF |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 6 | 8 | - |

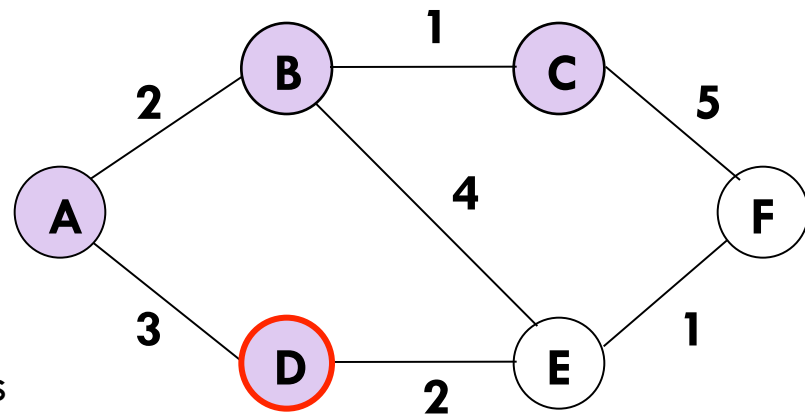
Caminhos

| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ABE | ABCF |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 5 | 8 | - |

Caminhos

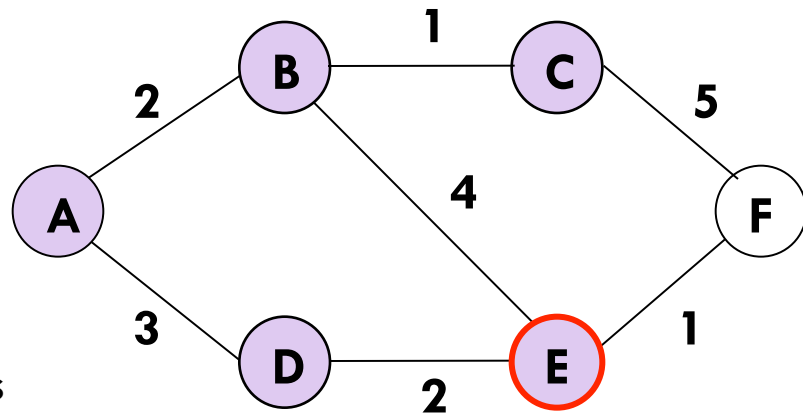
| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ADE | ABCF |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)



Distâncias

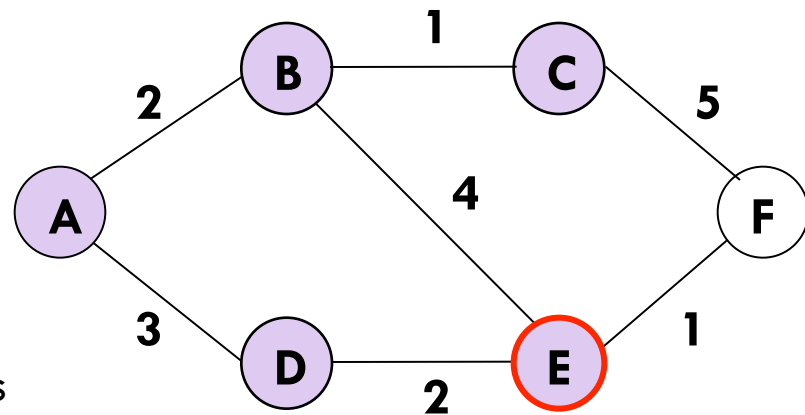
| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 5 | 8 | - |

Caminhos

| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ADE | ABCF |   |

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 5 | 8 | - |

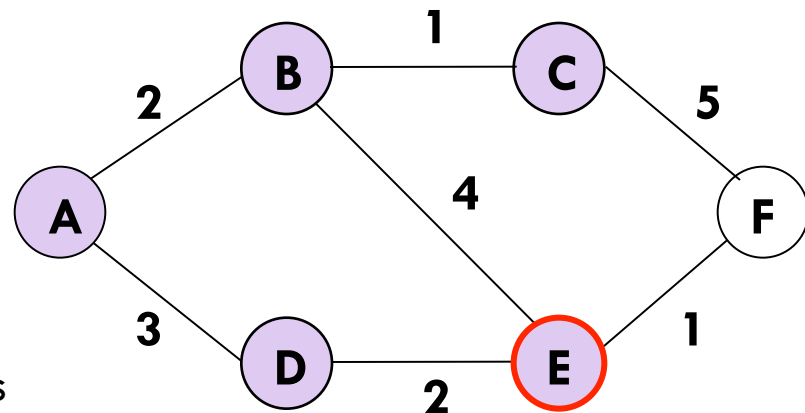
Caminhos

| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ADE | ABCF |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

G



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 5 | 6 | - |

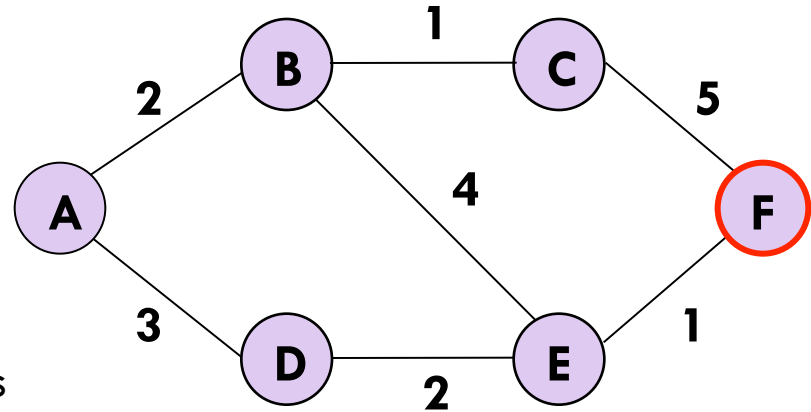
Caminhos

| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ADE | ADEF |   |

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)

# Algoritmo de Dijkstra

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)



Distâncias

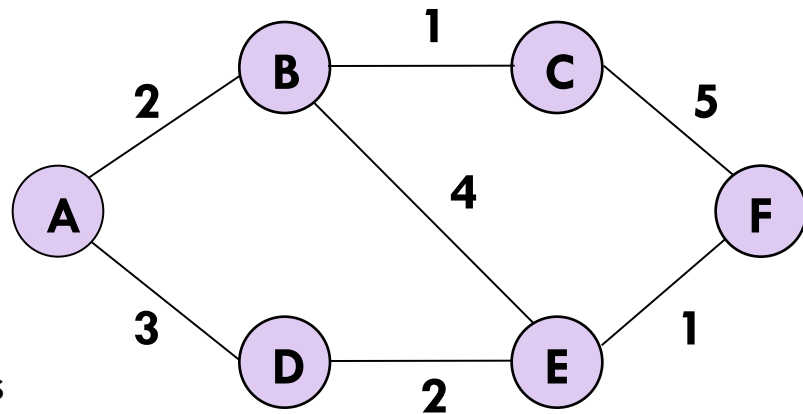
| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 5 | 6 | - |

Caminhos

| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ADE | ADEF |   |

# Algoritmo de Dijkstra

- (1) Inicializações
- (2) Escolher um vértice não visitado  $x$  cuja distância mínima para  $V_0$  seja a menor conhecida. Se  $x$  for NULO, termine o algoritmo
- (3) Marcar  $x$  como visitado
- (4) **Para** cada vizinho  $i$  não visitado de  $x$  se  $d(V_0, x) + \text{aresta}(x, i) < d(V_0, i)$   
**faça**
  - (i)  $d(V_0, i) = d(V_0, x) + \text{aresta}(x, i)$
  - (ii)  $c(i) = c(x) + i$
- (5) Se existirem vértices não visitados voltar para o passo (2)



Distâncias

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0 | 2 | 3 | 3 | 5 | 6 | - |

Caminhos

| A | B  | C   | D  | E   | F    | G |
|---|----|-----|----|-----|------|---|
| A | AB | ABC | AD | ADE | ADEF |   |