Floyd–Warshall e Dijkstra

Tópicos Avançados em Programação

Instituto de Informática
Universidade Federal de Goiás

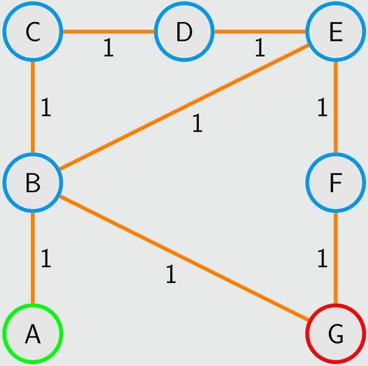
16 de Maio de 2018

Sumário

Introdução

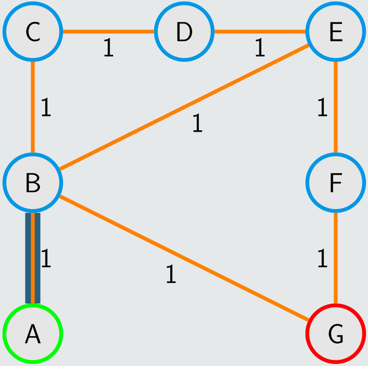
Caminho mínimo

Menor caminho de A para G. Custo das arestas $= 1$. BFS!



Introdução
Caminho mínimo

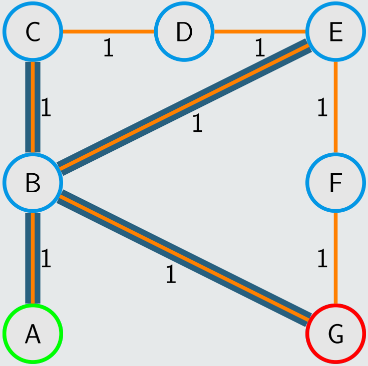
Menor caminho de A para G. Custo das arestas $= 1$. BFS!



Introdução

Caminho mínimo

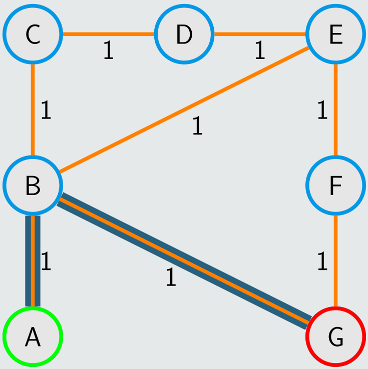
Menor caminho de A para G. Custo das arestas $= 1$. BFS!



Introdução

Caminho mínimo

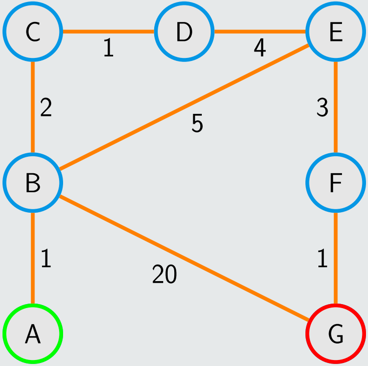
Menor caminho de A para G. Custo das arestas $= 1$. BFS!



Introdução

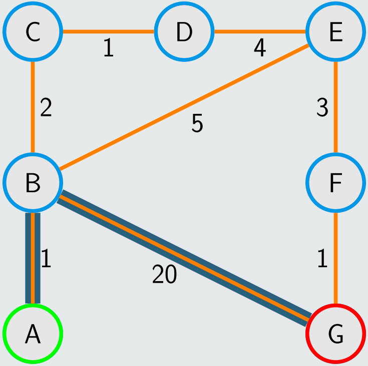
Caminho mínimo

Menor caminho de A para G. Custo das arestas $\neq 1$. BFS?



Introdução
Caminho mínimo

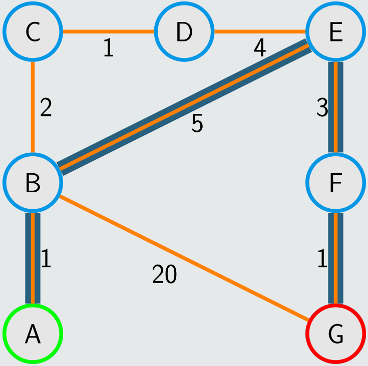
Menor caminho de A para G. Custo das arestas $\neq 1$. BFS?



Introdução

Caminho mínimo

Menor caminho de A para G. Custo das arestas $\neq 1$. BFS?



Introdução

Então como resolver?

✓ Dado um grafo $G = (V, E)$

- Quais são as menores distâncias entre cada par de vértices v_i e v_j, onde v_i, $v_j \in V$ e $N \leqslant 200$.
- Quais são as menores distâncias entre um vértice v_i para todos os outros vértices v_j, onde $v_i, v_j \in V$ e $N \leqslant 1000$.

Introdução

Então como resolver?

- ✓ Dado um grafo $G = (V, E)$
  - Quais são as menores distâncias entre cada par de vértices v_i e v_j, onde v_i, $v_j \in V$ e $N \leqslant 200$.
    - Algoritmo Flody-Warshall
    - Complexidade $\mathcal{O}(|V|^3)$
  - Quais são as menores distâncias entre um vértice v_i para todos os outros vértices v_j, onde $v_i, v_j \in V$ e $N \leqslant 1000$.

Floyd-warshall

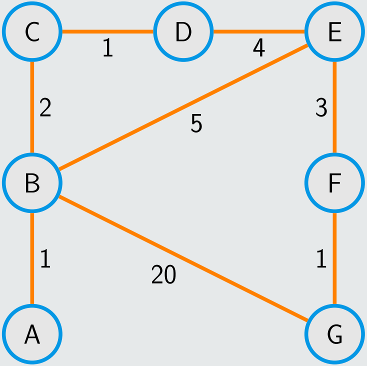
Rpresentação do grafo - Matriz de adjacência

✓ Dado um grafo $G = (V, E)$, a matriz de adjacência M é definida por uma matriz de ordem $|V|x|V|$, de forma que:

- $M[i, i] = 0$, $\forall i \in V$
- $M[i, j] = custo(i, j)$, se existir aresta de i para j
- $M[i, j] = \infty$, se não existir aresta de i para j

Floyd-warshall

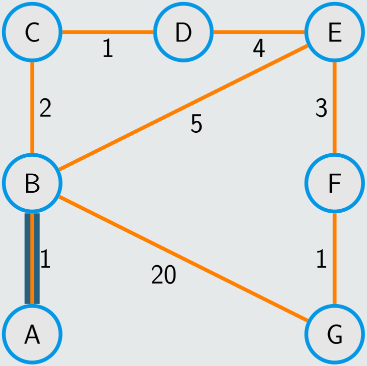
Representação do grafo - Matriz de adjacência



	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	∞
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

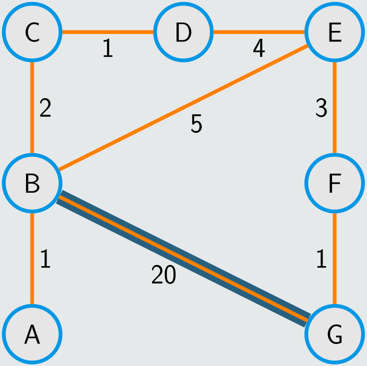
Representação do grafo - Matriz de adjacência



	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	∞
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Representação do grafo - Matriz de adjacência



	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	∞
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Vértice intermediário

✓ Um vértice intermediário em um caminho do vértice v_1 até v_n, presentado por $C = \{v_1,\ v_2,\ \ldots\ ,\ v_{n-1}, v_n\}$, é qualquer vértice u que não seja v_1 ou v_n, ou seja, $u \in V$-$\{v_1,\ v_n\}$.

$$v_1 \longrightarrow v_2 \dashrightarrow u \dashrightarrow v_{n-1} \longrightarrow v_n$$

Floyd-warshall

Ideia do algoritmo

✓ A ideia principal do algoritmo se baseia no principio de que um caminho entre dois vértices v_i e v_j existe se:

0: há uma aresta v_i para v_j, ou

1: há um caminho de v_i para v_j que passa por vértices intermediários do conjunto de vértices $k = \{v_1\}$, ou

2: há um caminho de v_i para v_j que passa por vértices intermediários do conjunto de vértices $k = \{v_1, v_2\}$, ou

⋮

n: há um caminho de v_i para v_j que passa por vértices intermediários do conjunto de vértices $k = \{ v_1, v_2, \ldots, v_{n-1}, v_n\}$.

Floyd-warshall

Algoritmo - Inicialização da matriz de adjacência

```
int grafo[n][n];

for(int i = 0; i < n; i++){
    for(int j = 0; j < n; j++){
        grafo[i][j] = 1e9;
    }
    grafo[i][i] = 0;
}
```

Floyd-warshall

Algoritmo - Preenchimento da matriz de adjacência

```
for(int i = 0; i < m; i++){
    //indexado de 0
    cin >> de >> para >> custo;
    grafo[ de ][ para ] = custo;
    //caso o grafo nao seja orientado
    grafo[ para ][ de ] = custo;
}
```

Floyd-warshall

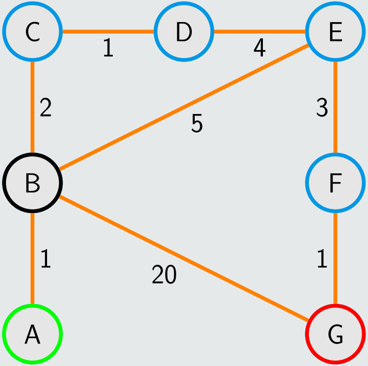
Algoritmo - Floyd-warshall

```
for(int k = 0; k < n; k++){
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            grafo[i][j] = min( grafo[i][j], grafo[i][k] + grafo[k][j] );
        }
    }
}
```

Floyd-warshall

Exemplo de execução

```
grafo[A][G] = min( grafo[A][G], grafo[A][B] + grafo[B][G] );
```

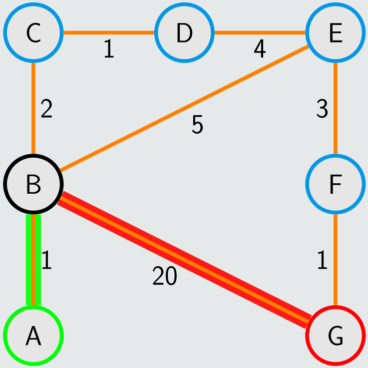


	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	∞
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][G] = min( 1e9, 1 + 20 );
```

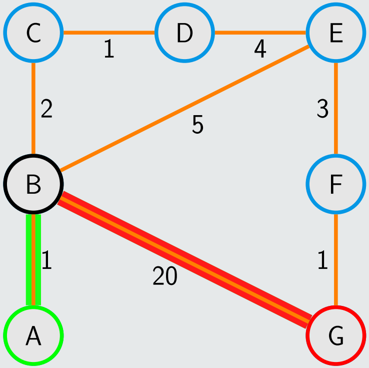


	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	∞
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][G] = 21;
```

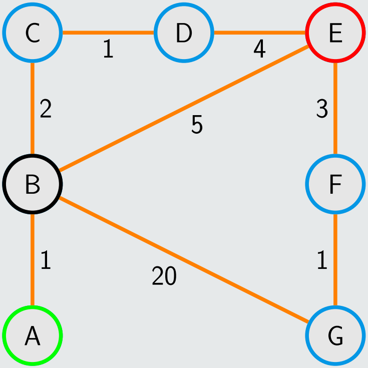


	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][E] = min( grafo[A][E], grafo[A][B] + grafo[B][E] );
```

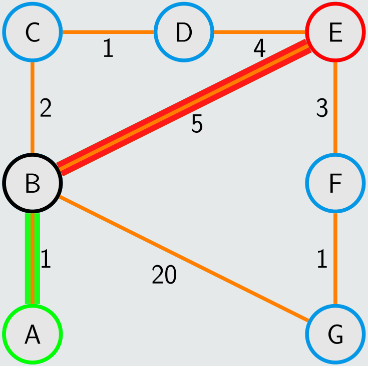


	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][E] = min( 1e9, 1 + 5 );
```

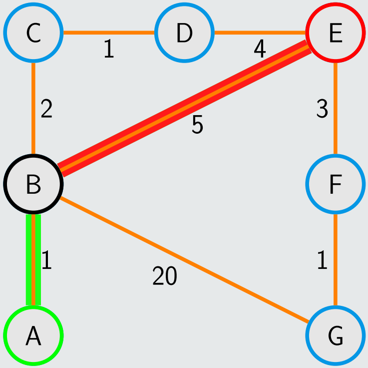


	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][E] = 6;
```

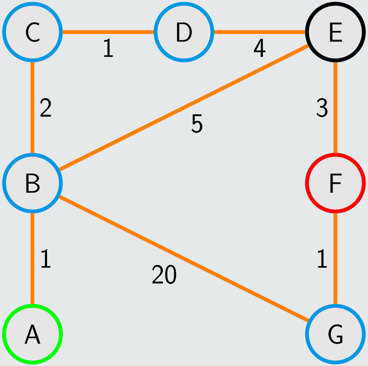


	A	B	C	D	E	F	G
A	0	1	∞	∞	6	∞	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][F] = min( grafo[A][F], grafo[A][E] + grafo[E][F] );
```

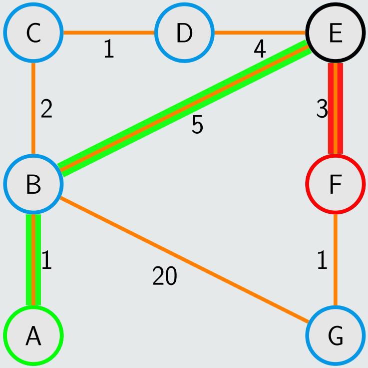


	A	B	C	D	E	F	G
A	0	1	∞	∞	6	∞	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][F] = min( 1e9, 6 + 3 );
```

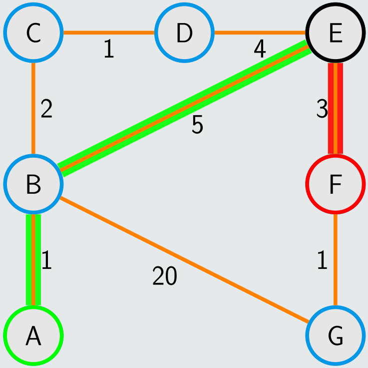


	A	B	C	D	E	F	G
A	0	1	∞	∞	6	∞	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][F] = 9;
```

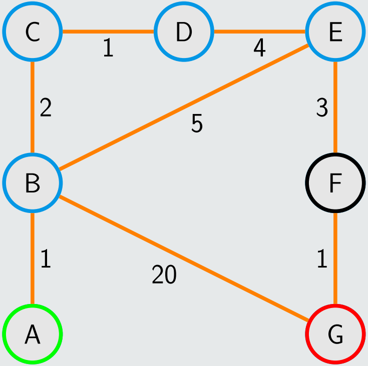


	A	B	C	D	E	F	G
A	0	1	∞	∞	6	9	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][G] = min( grafo[A][G], grafo[A][F] + grafo[F][G] );
```

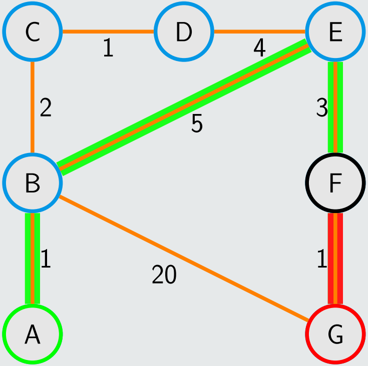


	A	B	C	D	E	F	G
A	0	1	∞	∞	6	9	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][G] = min( 21, 9 + 1 );
```

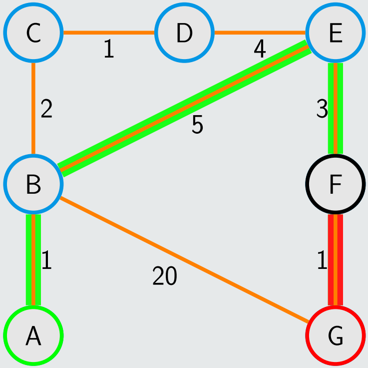


	A	B	C	D	E	F	G
A	0	1	∞	∞	6	9	21
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Floyd-warshall

Exemplo de execução

```
grafo[A][G] = 10;
```



	A	B	C	D	E	F	G
A	0	1	∞	∞	6	9	10
B	1	0	2	∞	5	∞	20
C	∞	2	0	1	∞	∞	∞
D	∞	∞	1	0	4	∞	∞
E	∞	5	∞	4	0	3	∞
F	∞	∞	∞	∞	3	0	1
G	∞	20	∞	∞	∞	1	0

Introdução

Então como resolver?

- ✓ Dado um grafo $G = (V, E)$
  - Quais são as menores distâncias entre cada par de vértices v_i e v_j, onde v_i, $v_j \in V$ e $N \leqslant 200$.
    - Algoritmo Flody-Warshall
    - Complexidade $\mathcal{O}(|V|^3)$
  - Quais são as menores distâncias entre um vértice v_i para todos os outros vértices v_j, onde $v_i, v_j \in V$ e $N \leqslant 1000$.

Introdução

Então como resolver?

- ✓ Dado um grafo $G = (V, E)$
  - Quais são as menores distâncias entre cada par de vértices v_i e v_j, onde v_i, $v_j \in V$ e $N \leqslant 200$.
    - Algoritmo Flody-Warshall
    - Complexidade $\mathcal{O}(|V|^3)$
  - Quais são as menores distâncias entre um vértice v_i para todos os outros vértices v_j, onde $v_i, v_j \in V$ e $N \leqslant 1000$.
    - Algoritmo Dijkstra
    - Complexidade $\mathcal{O}(|E| \log |V|)$

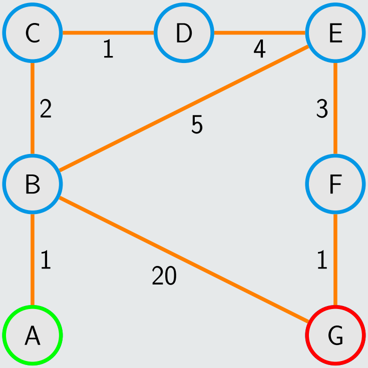
Dijkstra

Algoritmo - Dijkstra

✓ Exemplo de implementação : Dijkstra C++

Dijkstra

Exemplo de execução - Menor caminho de A para G



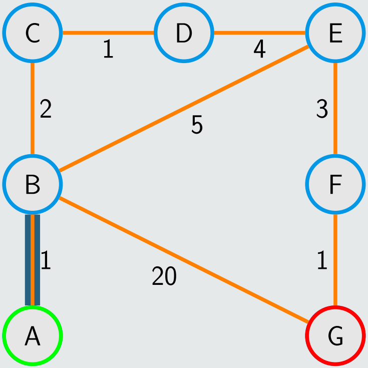
1

Dist	A	B	C	D	E	F	G
A	0	∞	∞	∞	∞	∞	∞

Fila	$0,0$

Dijkstra

Exemplo de execução - Menor caminho de A para G



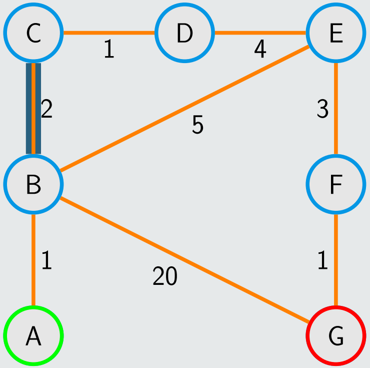
1

Dist	A	B	C	D	E	F	G
A	0	1	∞	∞	∞	∞	∞

Fila	$0, A$	$1, B$

Dijkstra

Exemplo de execução - Menor caminho de A para G



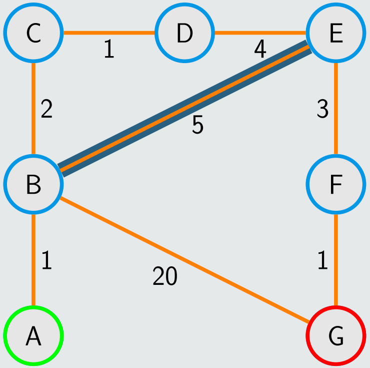
1

Dist	A	B	C	D	E	F	G
A	0	1	3	∞	∞	∞	∞

Fila	$1, B$	$3, C$

Dijkstra

Exemplo de execução - Menor caminho de A para G



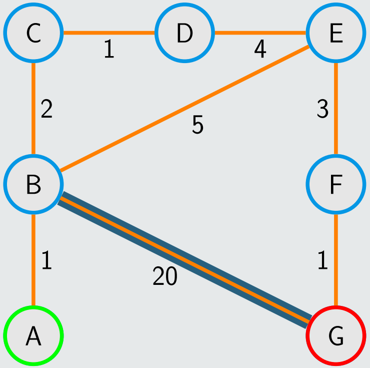
1

Dist	A	B	C	D	E	F	G
A	0	1	3	∞	6	∞	∞

Fila	$3, C$	$6, E$

Dijkstra

Exemplo de execução - Menor caminho de A para G



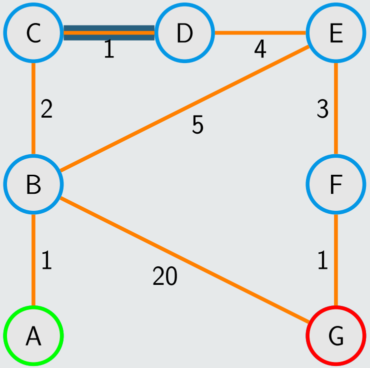
1

Dist	A	B	C	D	E	F	G
A	0	1	3	∞	6	∞	21

Fila	$1, B$	$3, C$	$6, E$	$21, G$

Dijkstra

Exemplo de execução - Menor caminho de A para G



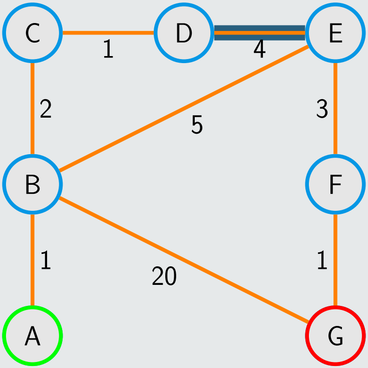
1

Dist	A	B	C	D	E	F	G
A	0	1	3	4	6	∞	20

Fila	$3, C$	$4, D$	$6, E$	$21, G$

Dijkstra

Exemplo de execução - Menor caminho de A para G



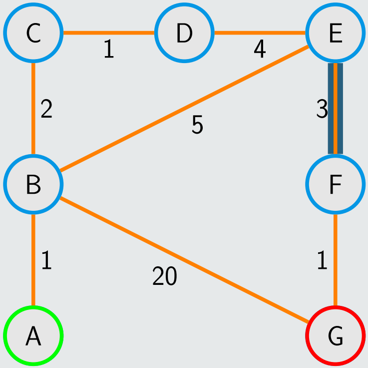
1

Dist	A	B	C	D	E	F	G
A	0	1	3	4	6	∞	21

Fila	$4, D$	$6, E$	$21, G$

Dijkstra

Exemplo de execução - Menor caminho de A para G



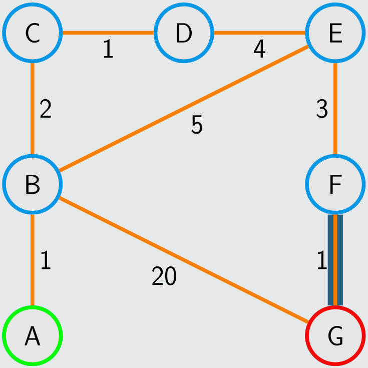
1

Dist	A	B	C	D	E	F	G
A	0	1	3	4	6	9	21

Fila	$6, E$	$9, F$	$21, G$

Dijkstra

Exemplo de execução - Menor caminho de A para G



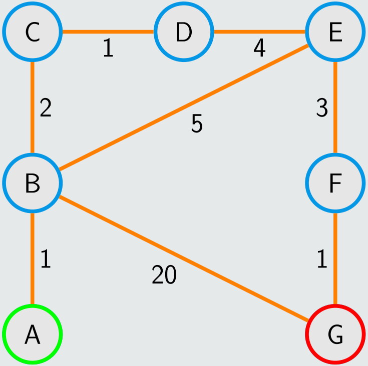
1

Dist	A	B	C	D	E	F	G
A	0	1	3	4	6	9	10

Fila	$9, F$	$10, G$	$21, G$

Dijkstra

Exemplo de execução - Menor caminho de A para G



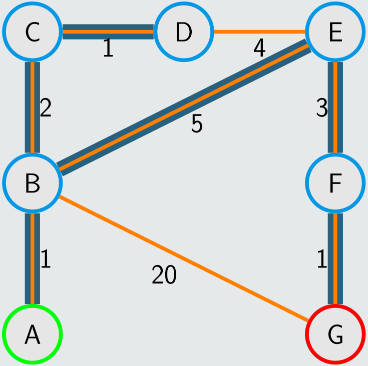
1

Dist	A	B	C	D	E	F	G
A	0	1	3	4	6	9	10

Fila	$10, G$	$21, G$

Dijkstra

Exemplo de execução - Menor caminho de A para G



1

Dist	A	B	C	D	E	F	G
A	0	1	3	4	6	9	10

Fila	$21, G$

Dúvidas

Dúvidas ?