MC-202 — Aula 25
Algoritmos em grafo

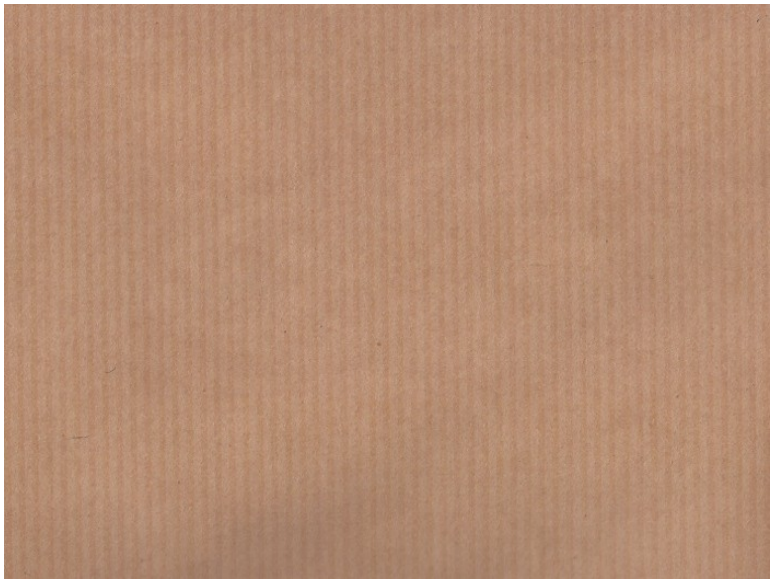
Lehilton Pedrosa

Instituto de Computação – Unicamp

Segundo Semestre de 2015

Roteiro

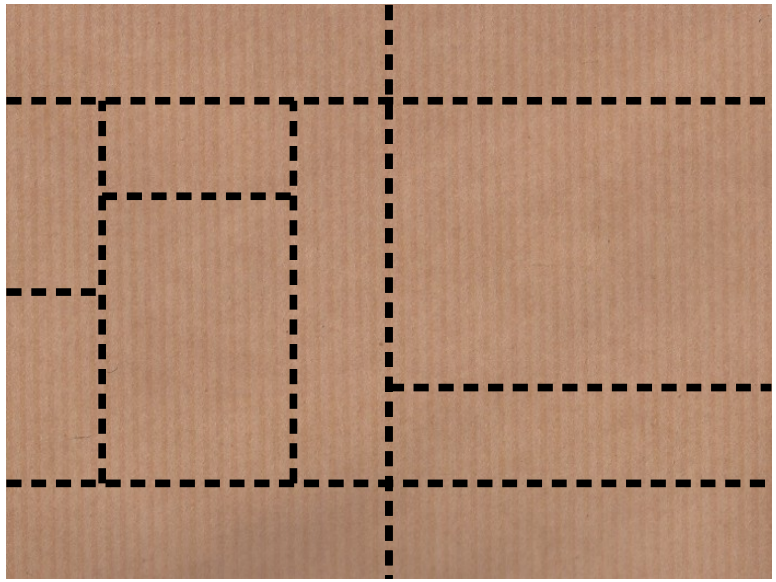
Corte de material



Problema

Uma fábrica utiliza como matéria prima um papelão retangular. Para criar certos objetos, o papel é cortado usando uma grande guilhotina (maior que o papelão).

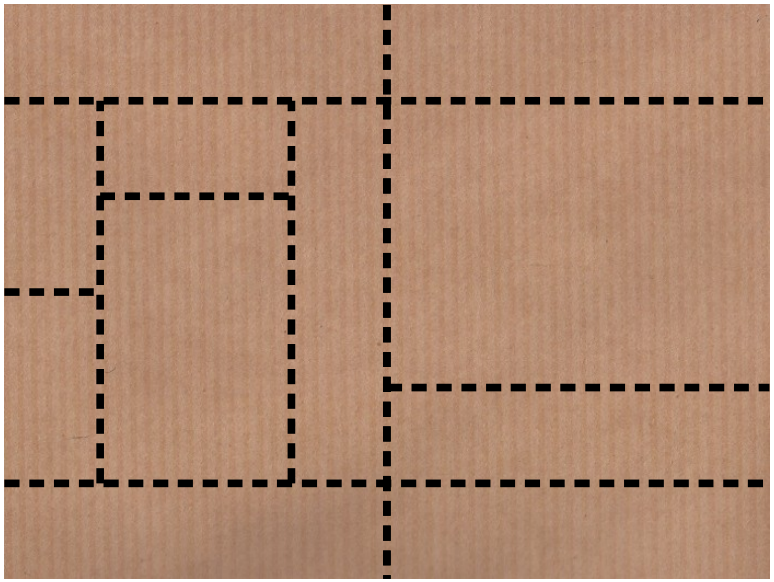
Corte de material



Problema

Uma fábrica utiliza como matéria prima um papelão retangular. Para criar certos objetos, o papel é cortado usando uma grande guilhotina (maior que o papelão). Para evitar o desperdício, um padrão de corte é cuidadosamente planejado, para que todo (ou quase todo o material seja utilizado).

Corte de material

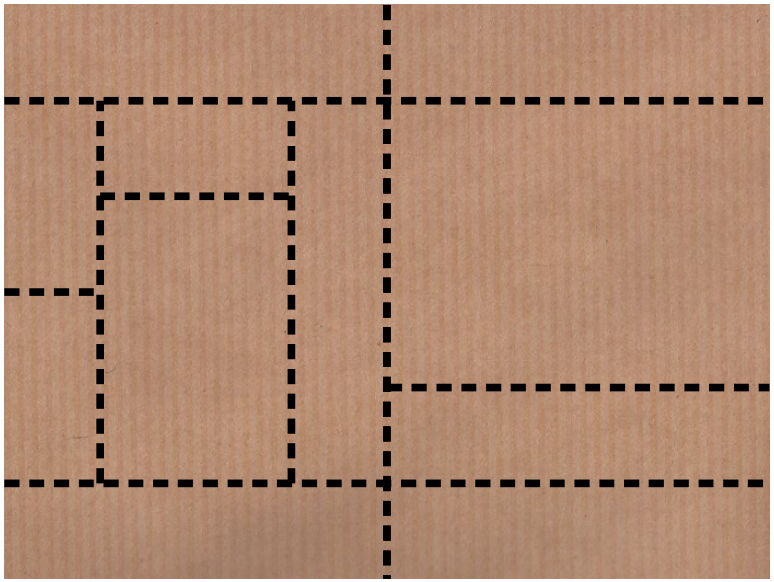


Problema

Uma fábrica utiliza como matéria prima um papelão retangular. Para criar certos objetos, o papel é cortado usando uma grande guilhotina (maior que o papelão). Para evitar o desperdício, um padrão de corte é cuidadosamente planejado, para que todo (ou quase todo o material seja utilizado).

1. Como verificar se o corte pode ser feito pela guilhotina?

Corte de material

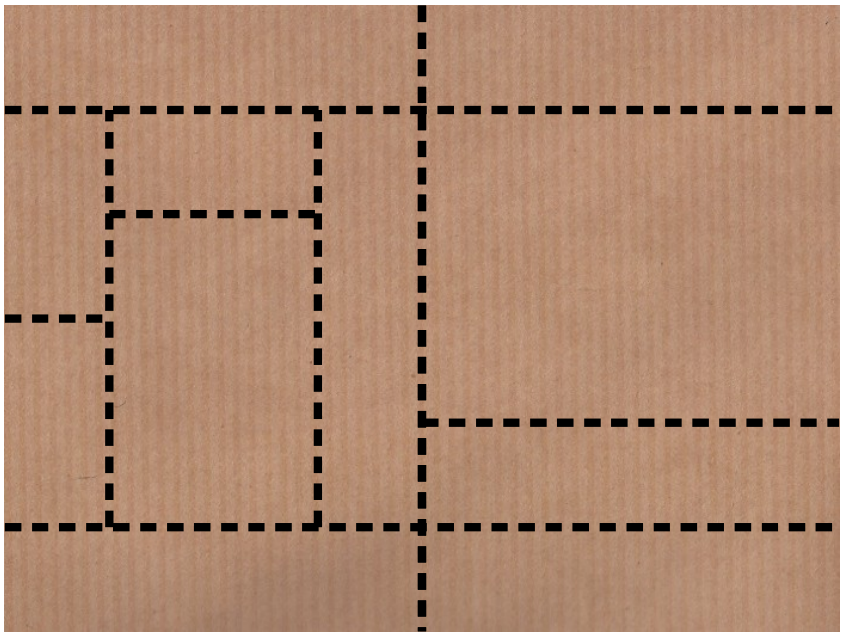


Problema

Uma fábrica utiliza como matéria prima um papelão retangular. Para criar certos objetos, o papel é cortado usando uma grande guilhotina (maior que o papelão). Para evitar o desperdício, um padrão de corte é cuidadosamente planejado, para que todo (ou quase todo o material seja utilizado).

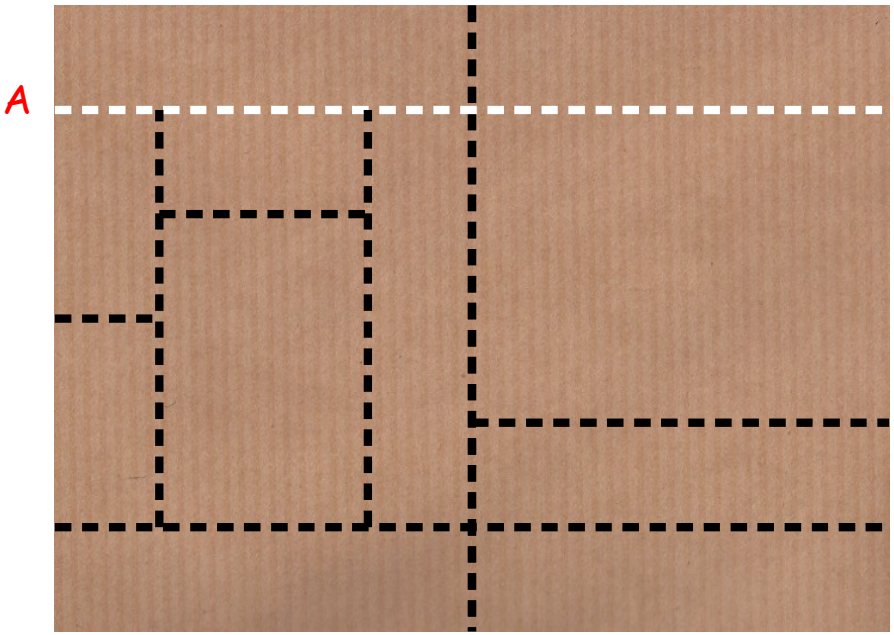
1. Como verificar se o corte pode ser feito pela guilhotina?
2. Como cortar o papel?

Modelando como um grafo



Modelando

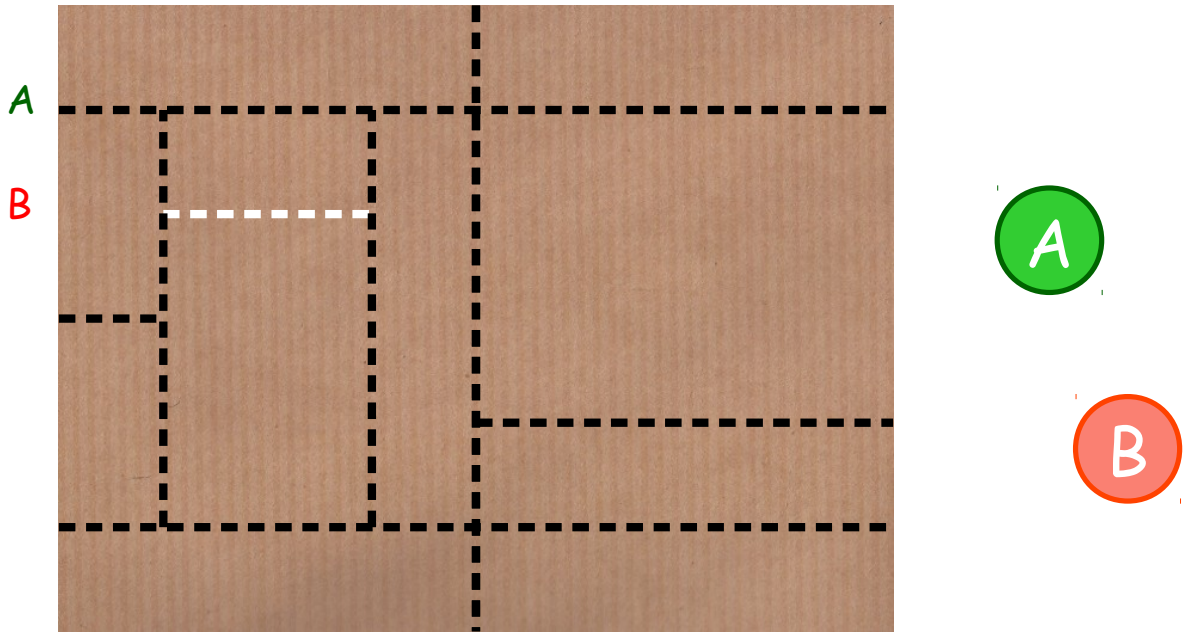
Modelando como um grafo



Modelando

- **Vértices:** consideramos cada corte da guilhotina

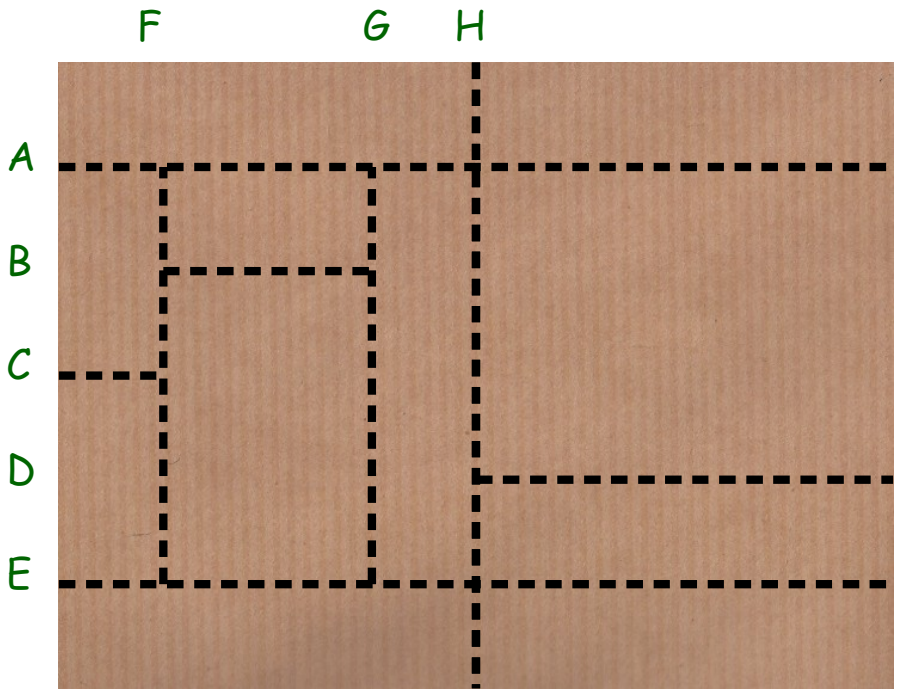
Modelando como um grafo

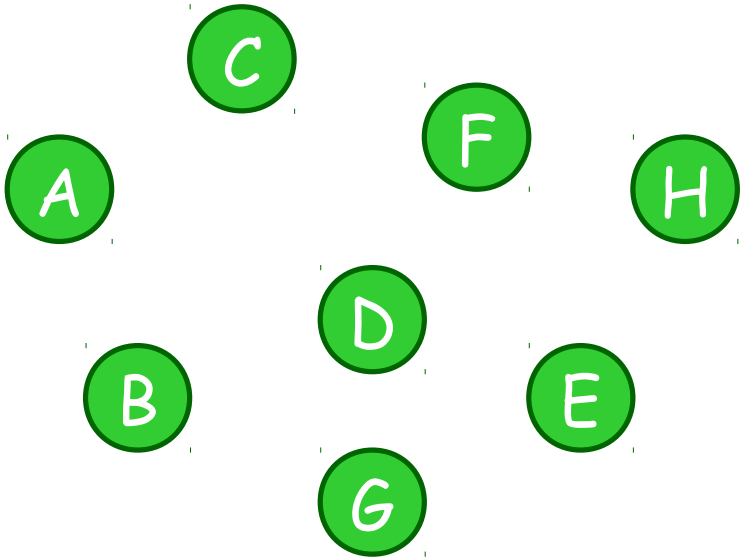


Modelando

- **Vértices:** consideramos cada corte da guilhotina

Modelando como um grafo

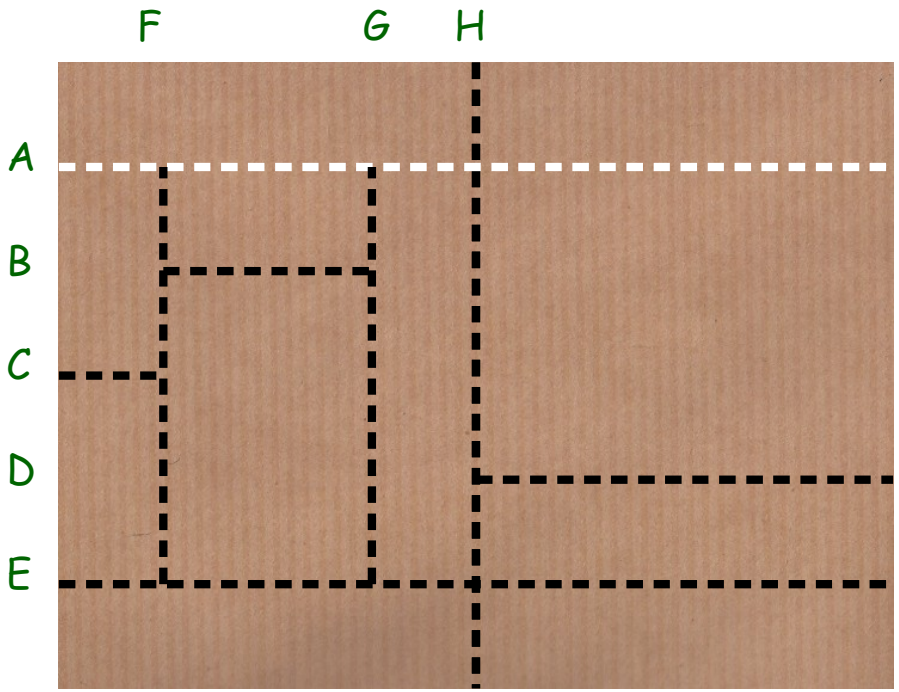


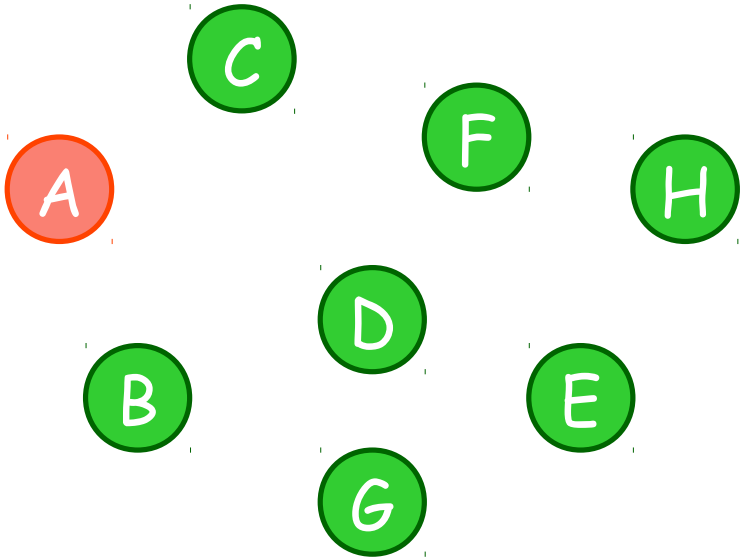


Modelando

- **Vértices:** consideramos cada corte da guilhotina

Modelando como um grafo

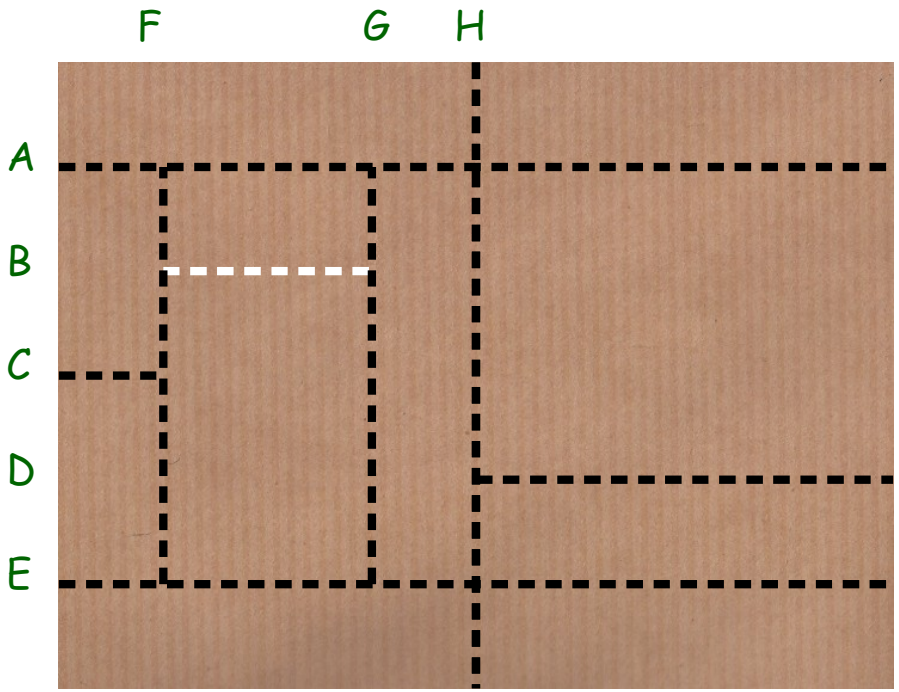


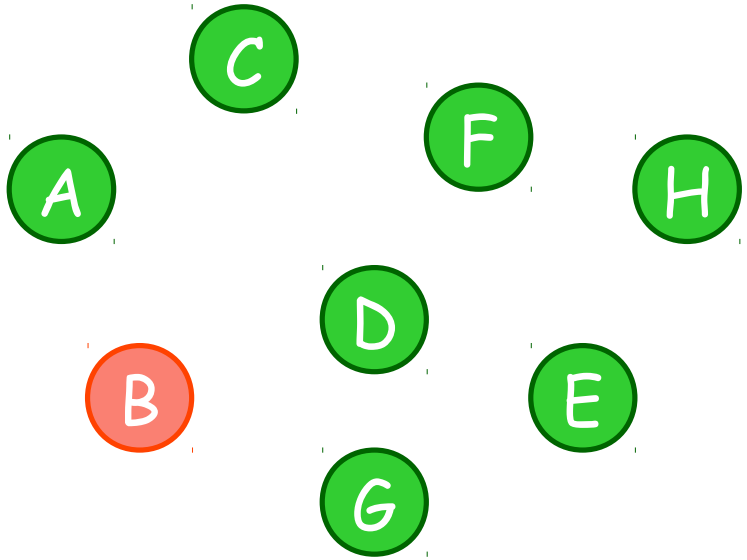


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo

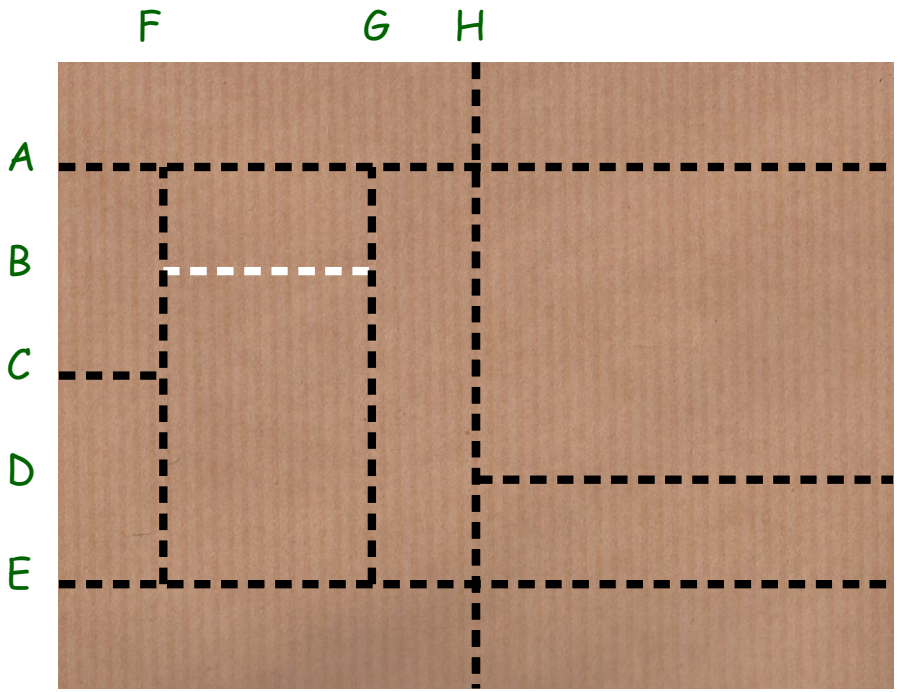


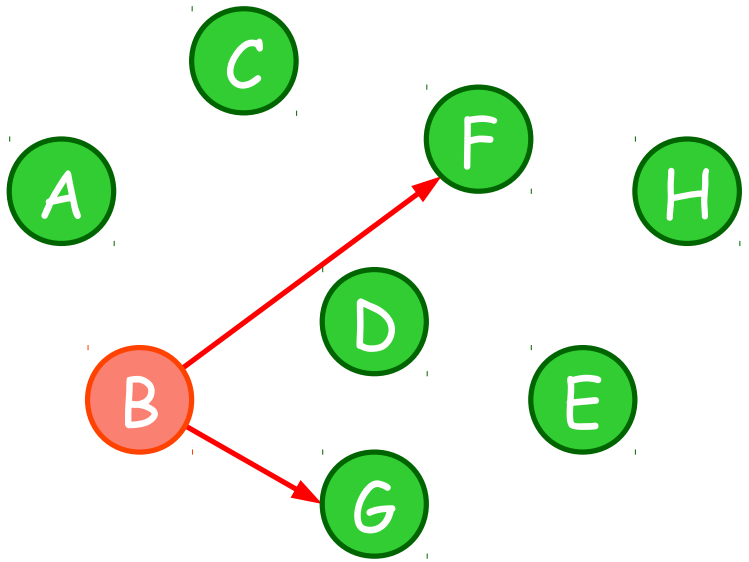


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo

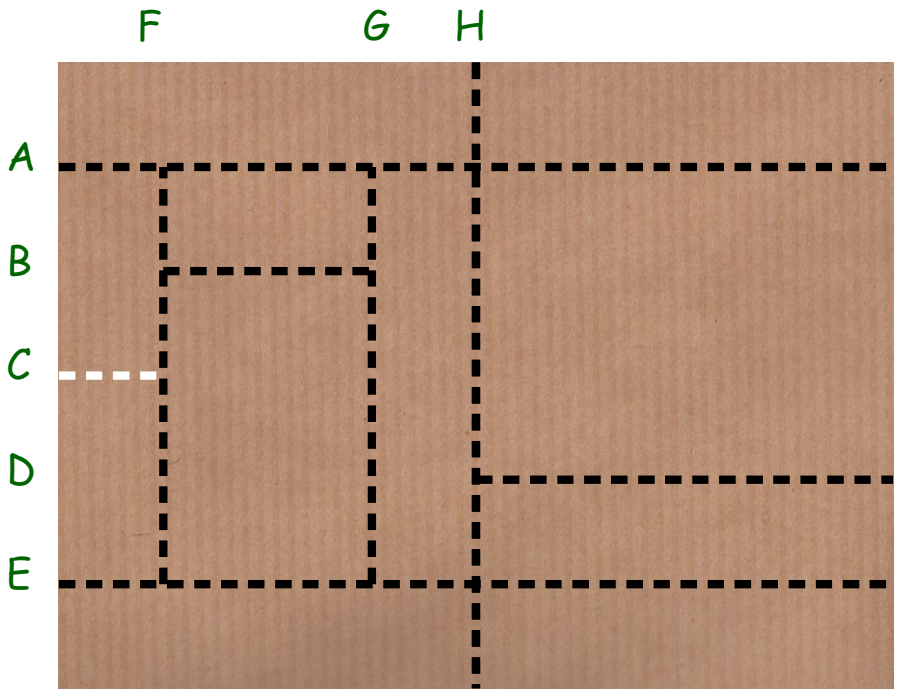


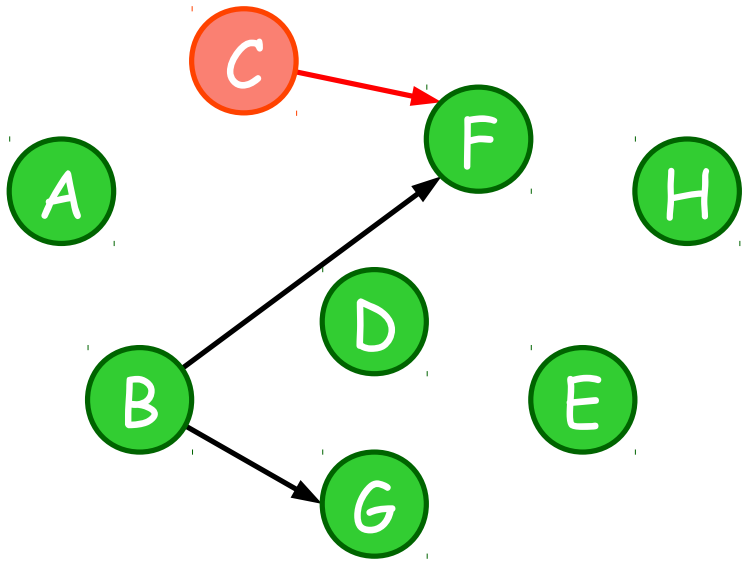


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo

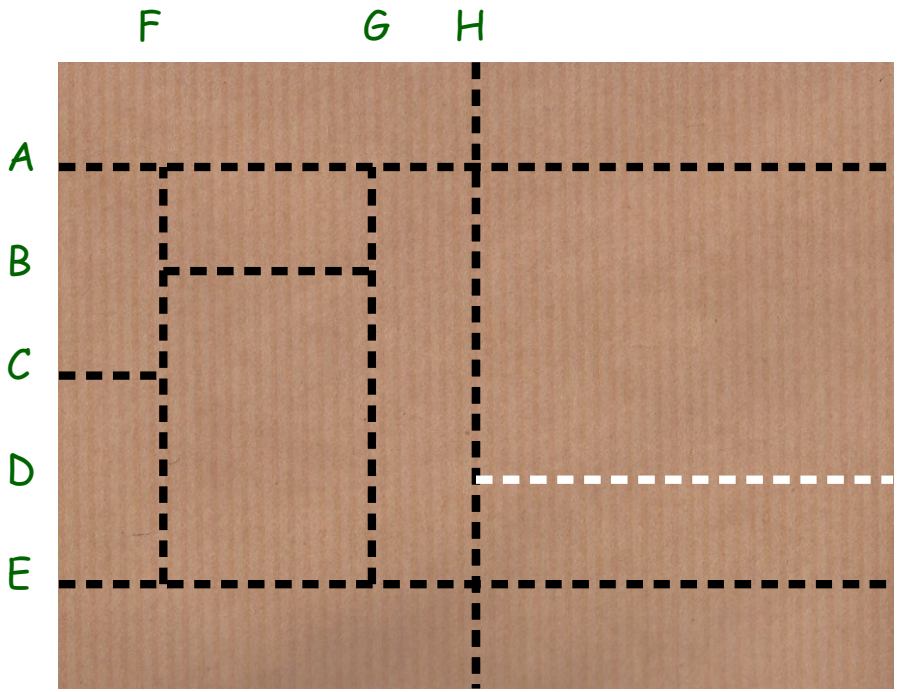


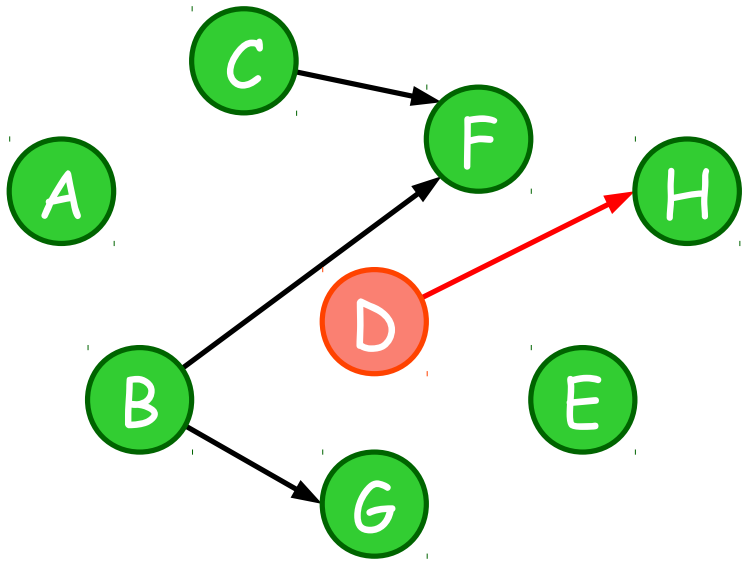


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo

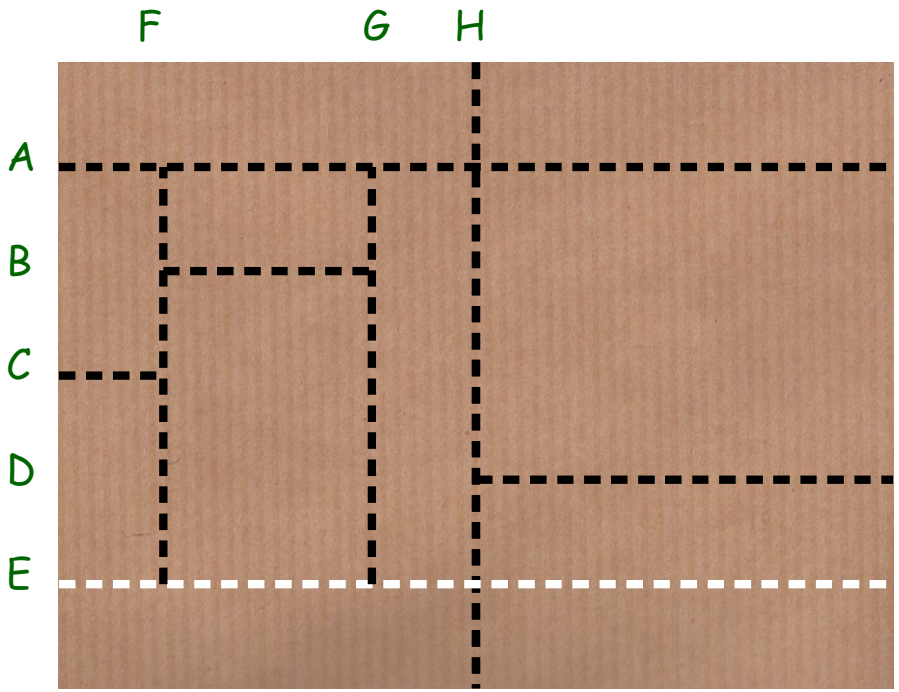


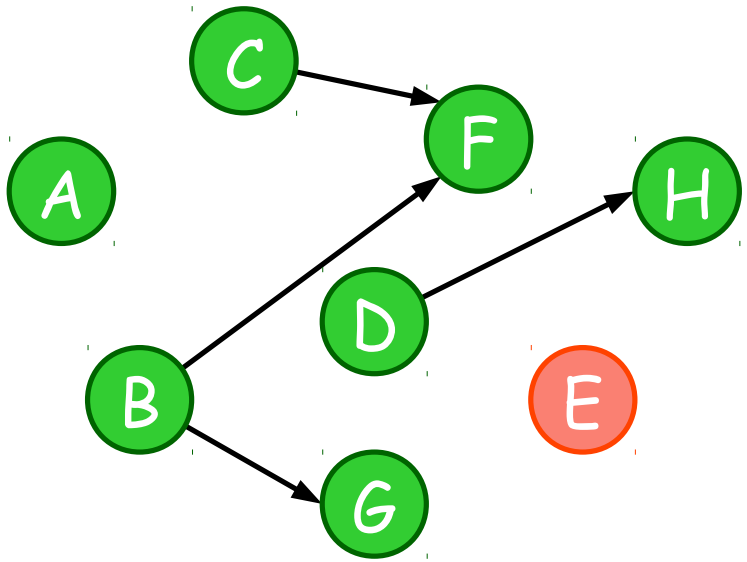


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo

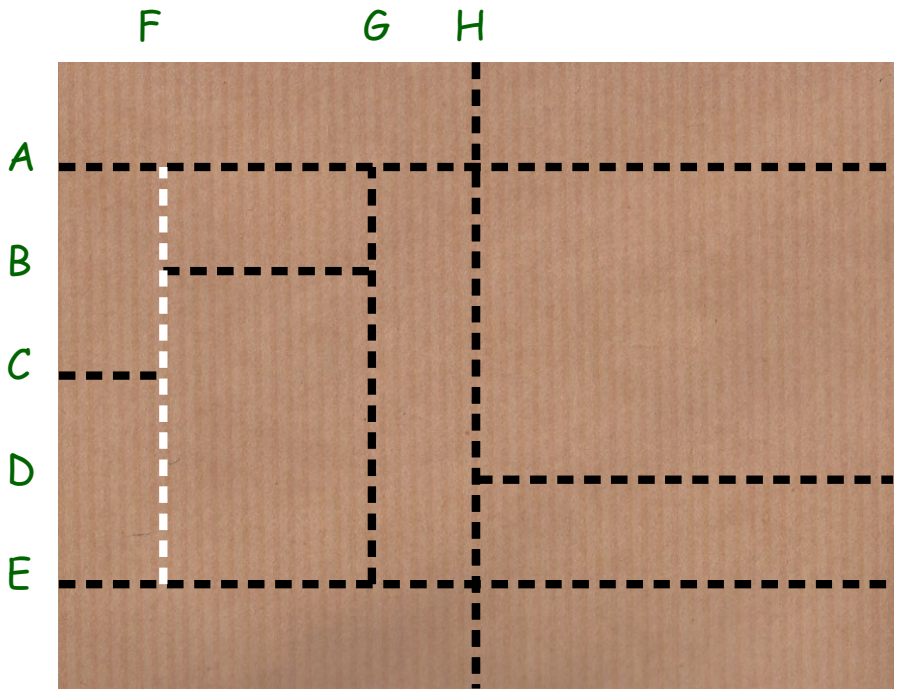


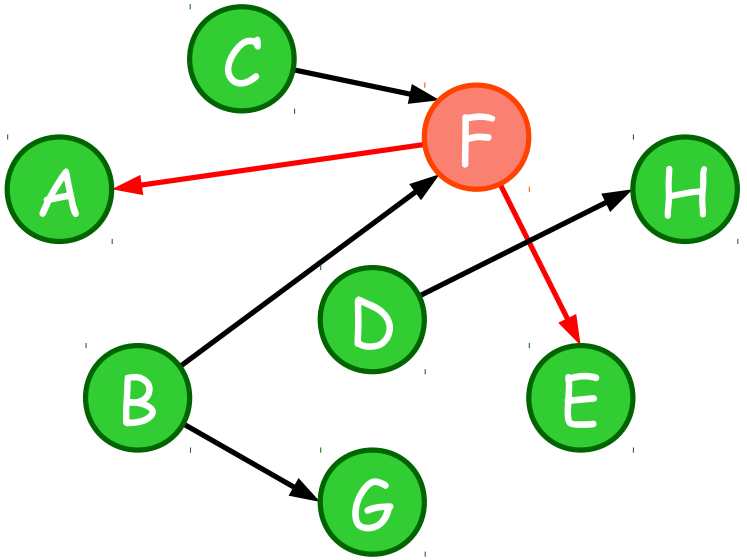


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo

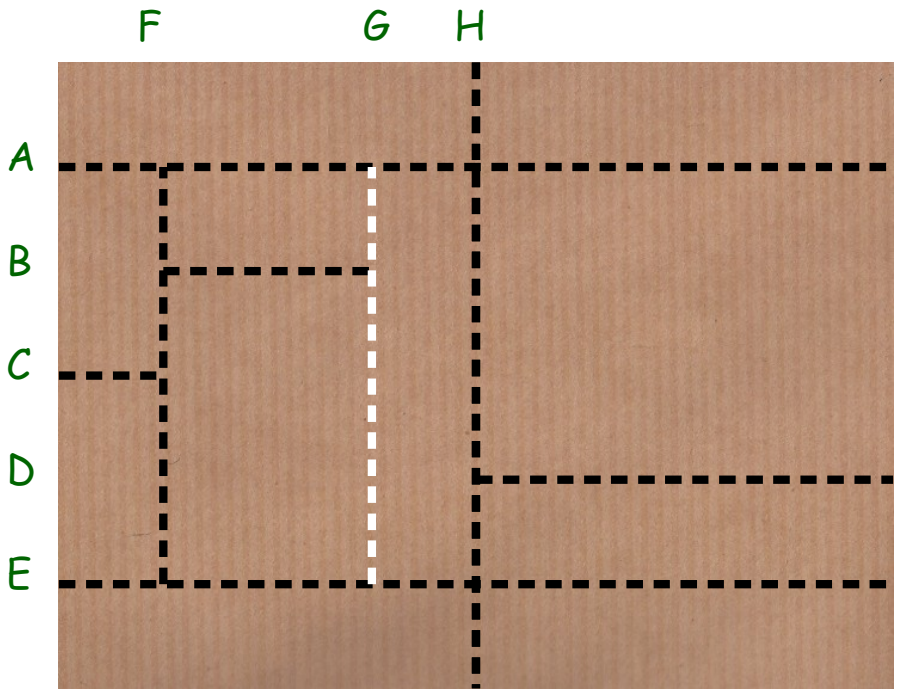


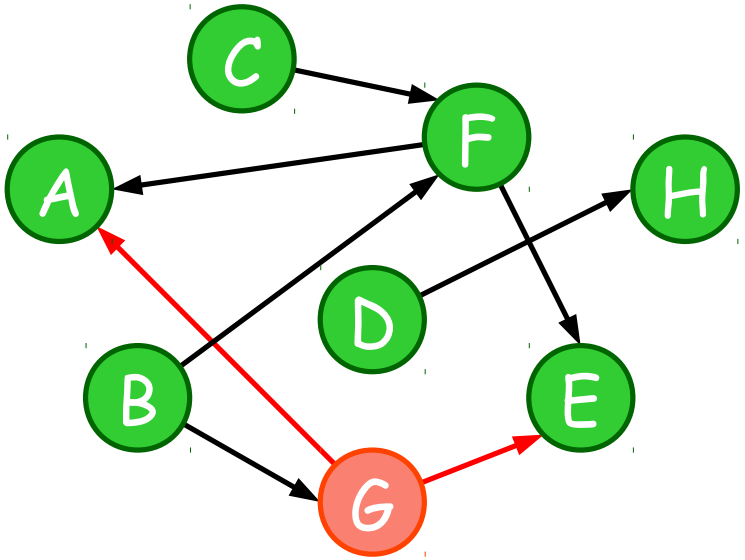


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo

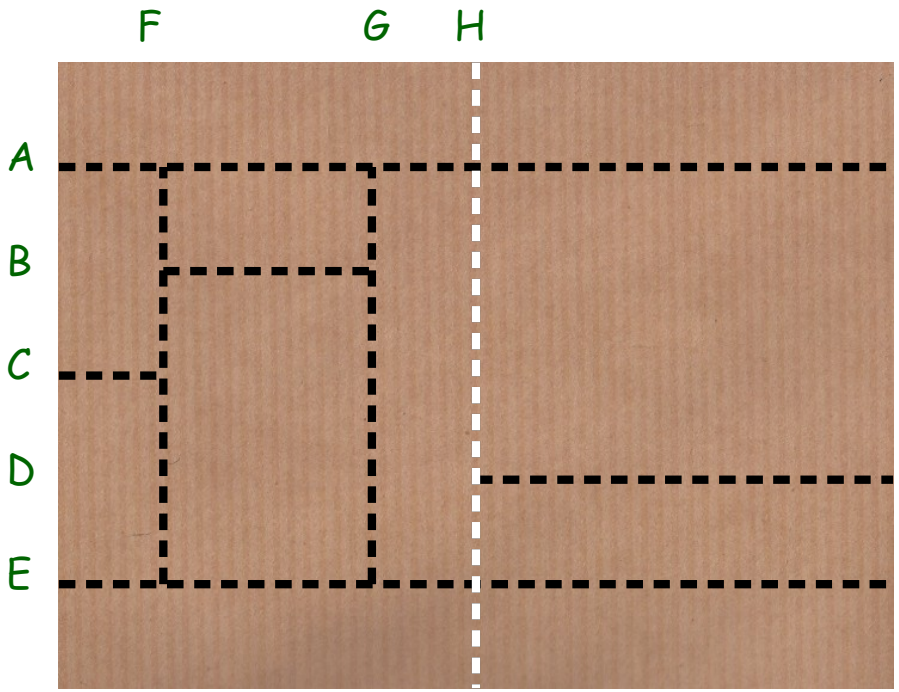


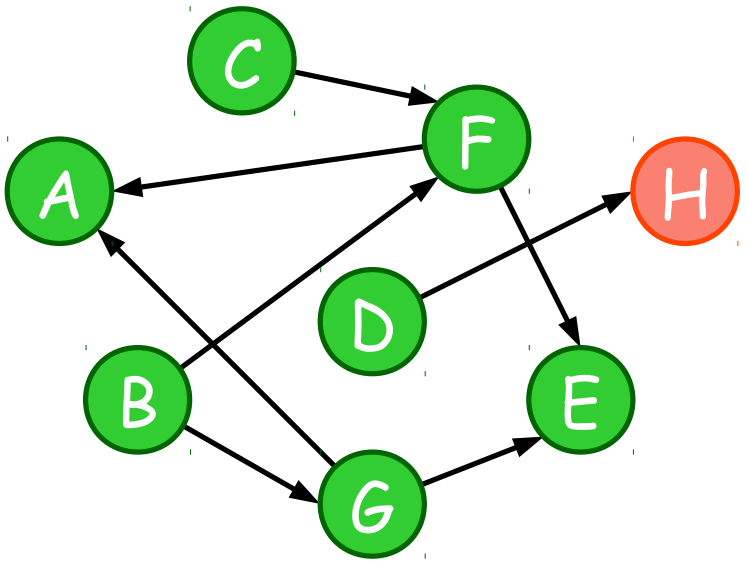


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo

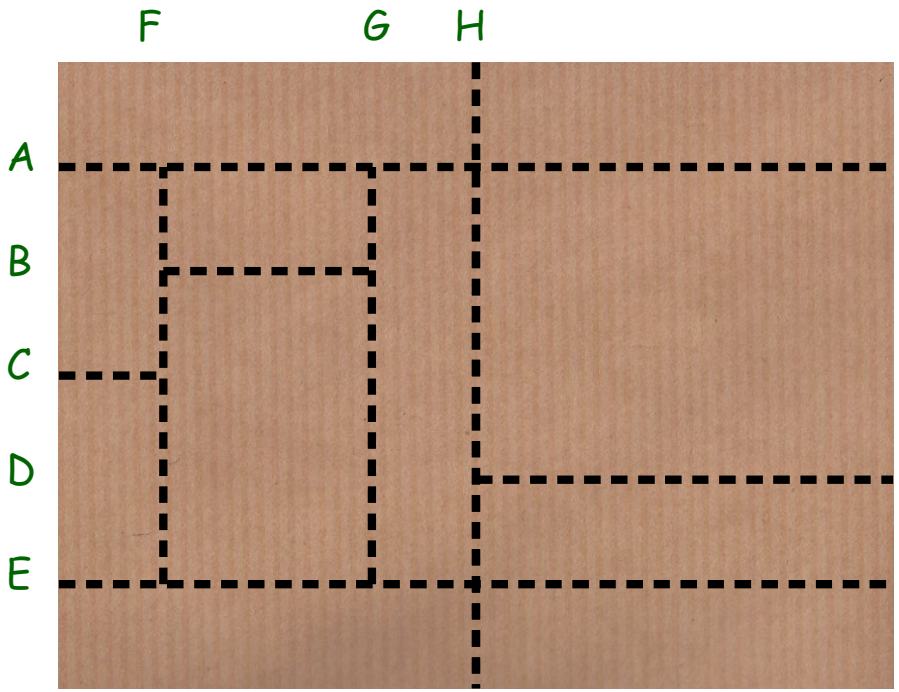


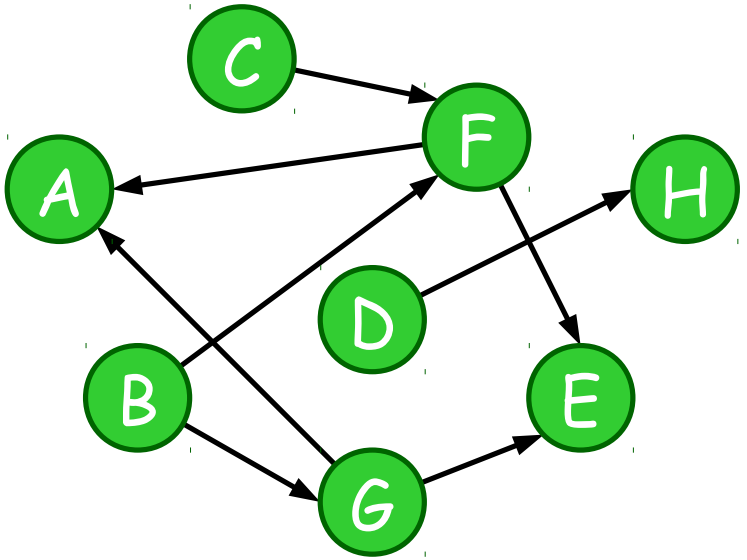


Modelando

- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Modelando como um grafo





Modelando

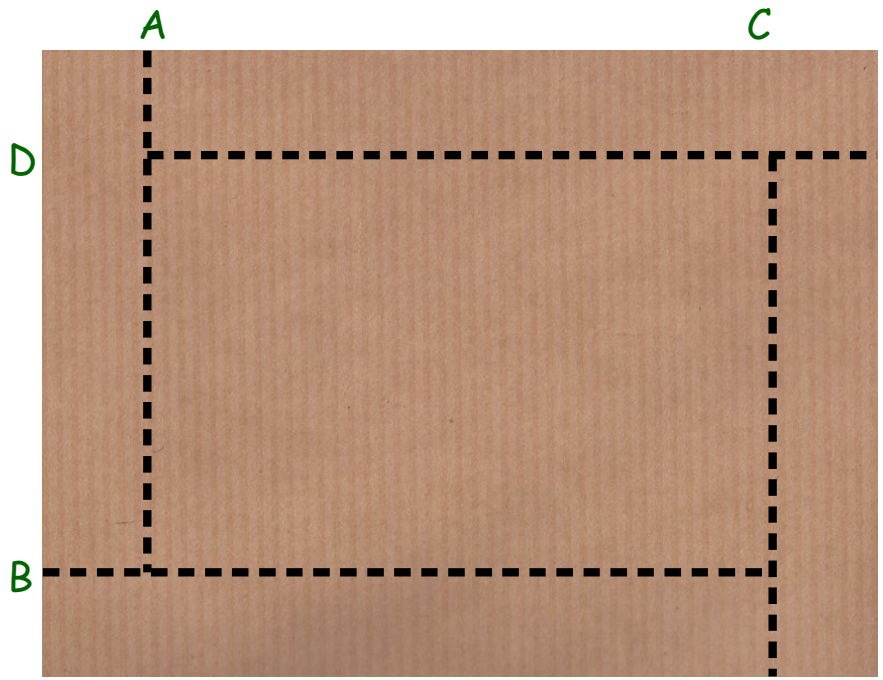
- **Vértices:** consideramos cada corte da guilhotina
- **Arestas:** qual a ordem entre um par de cortes

Viabilidade do padrão

Pergunta: Sempre é possível cortar um padrão com a guilhotina?

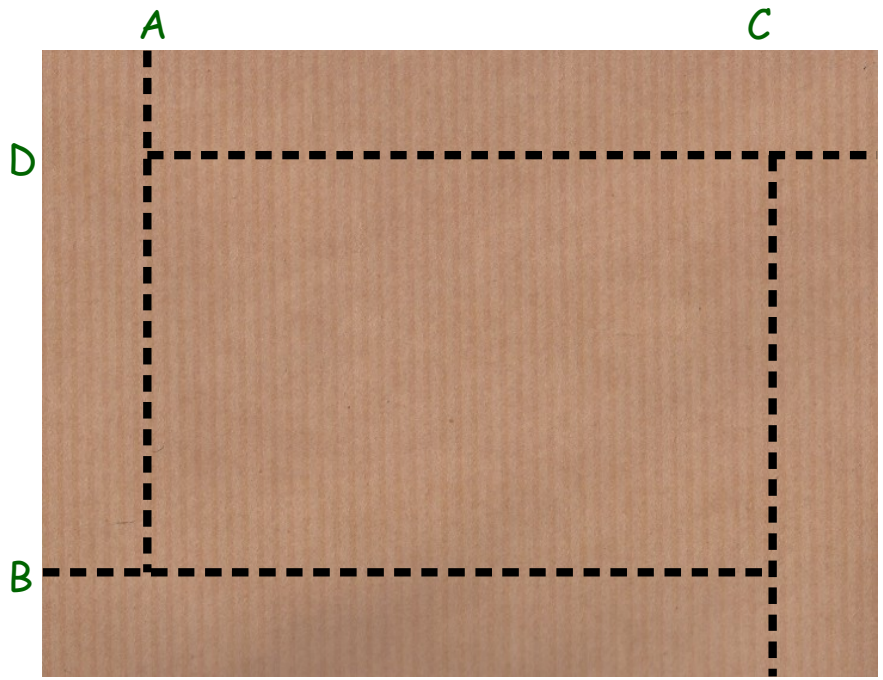
Viabilidade do padrão

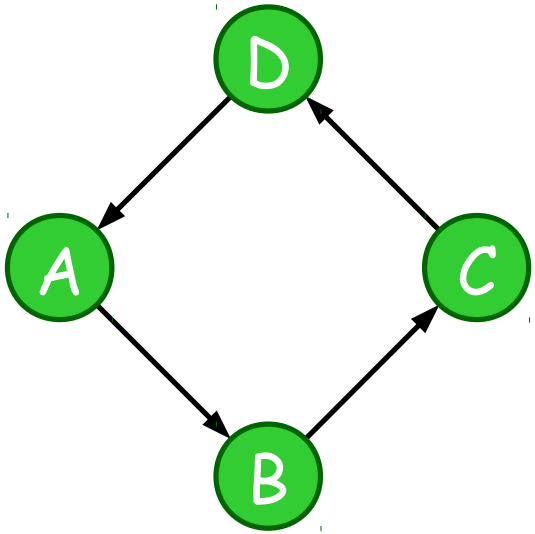
Pergunta: Sempre é possível cortar um padrão com a guilhotina?



Viabilidade do padrão

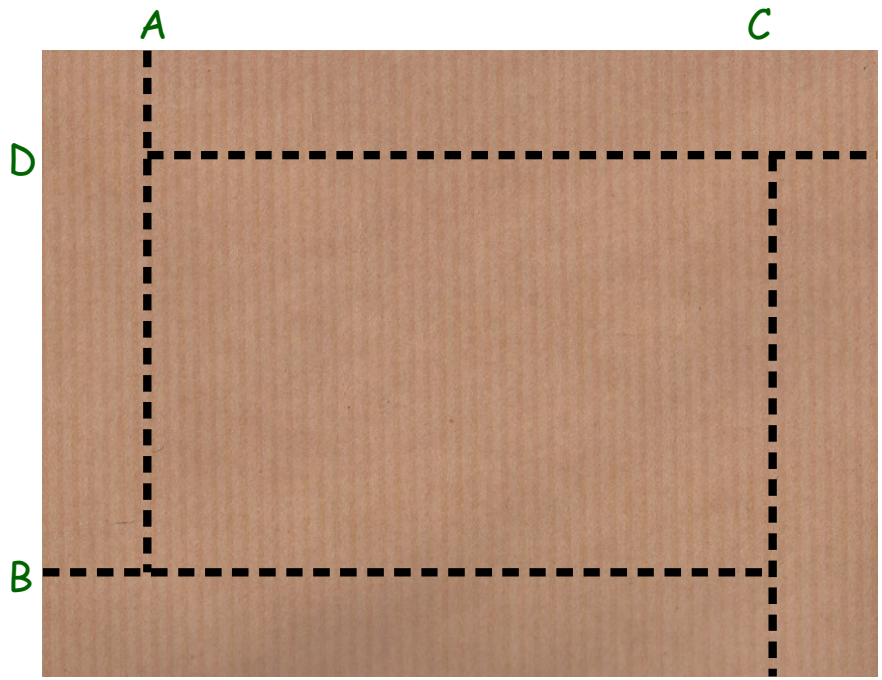
Pergunta: Sempre é possível cortar um padrão com a guilhotina?

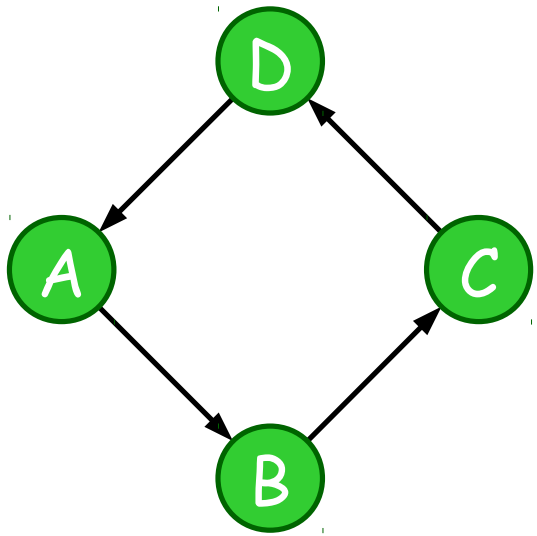




Viabilidade do padrão

Pergunta: Sempre é possível cortar um padrão com a guilhotina?





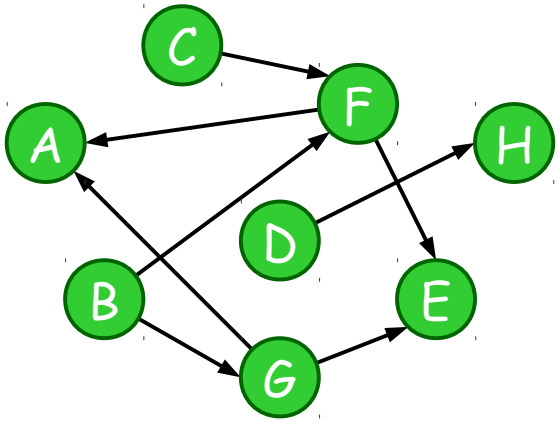
Conclusão

Um padrão de corte é viável se, e somente se, o grafo dos cortes for acíclico.

Ordenação topológica

Ordenação topológica

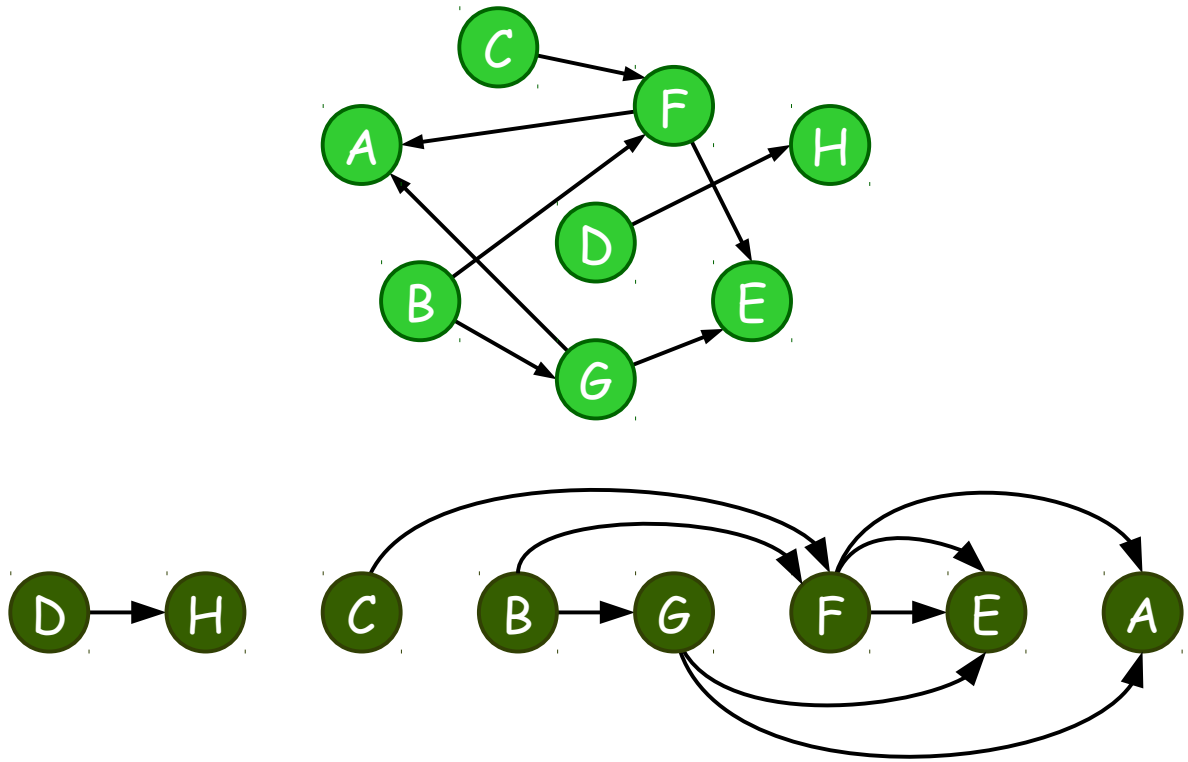
Dado um grafo acíclico,



Ordenação topológica

Ordenação topológica

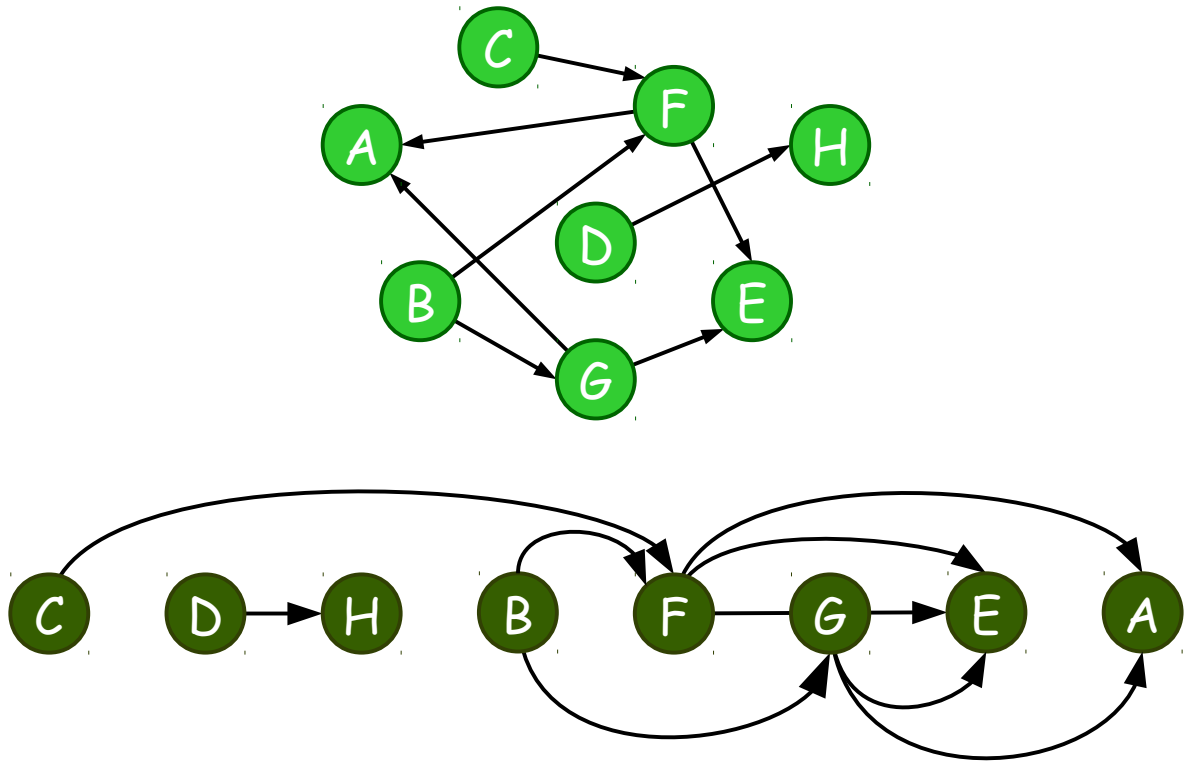
Dado um grafo acíclico, queremos obter uma ordenação dos vértices com arestas para frente.



Ordenação topológica

Ordenação topológica

Dado um grafo acíclico, queremos obter uma ordenação dos vértices com arestas para frente.



Observação: Pode haver várias ordenações possíveis.

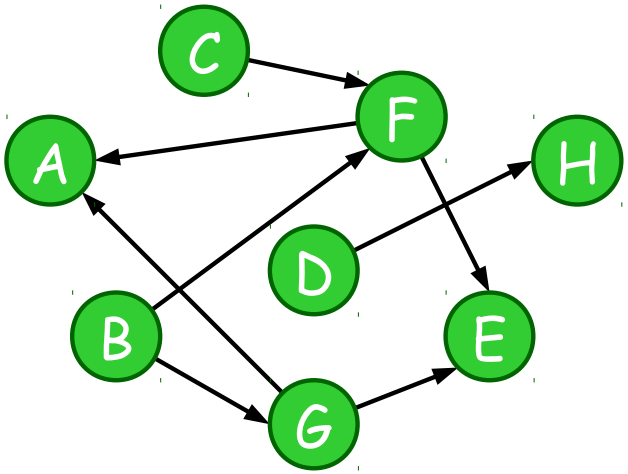
Ordenação topológica: algoritmo

Para obter uma ordenação, basta realizar uma busca em profundidade:

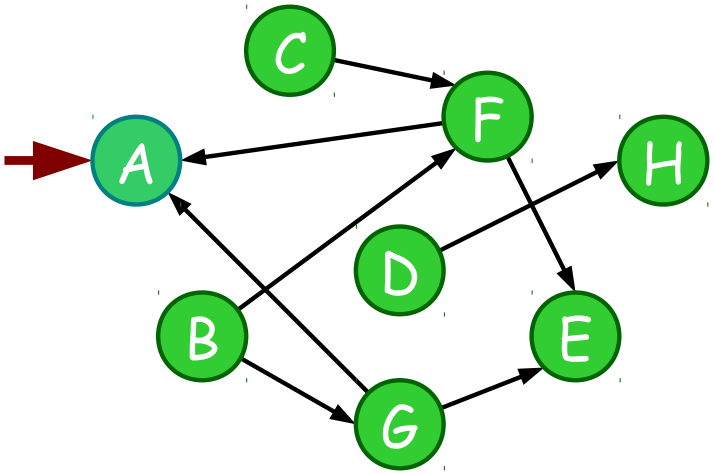
Algoritmo

1. Cada vértice mantém um estado:
   - não descoberto: não descoberto pela busca ainda
   - descoberto: já descoberto pela busca
   - terminado: já adicionada a lista de saída
2. Para cada vértice v não descoberto ainda:
   - marca v como visitado
   - visita os vizinho de v ainda não descobertos
   - insere v no início da lista

Ordenação topológica: exemplo

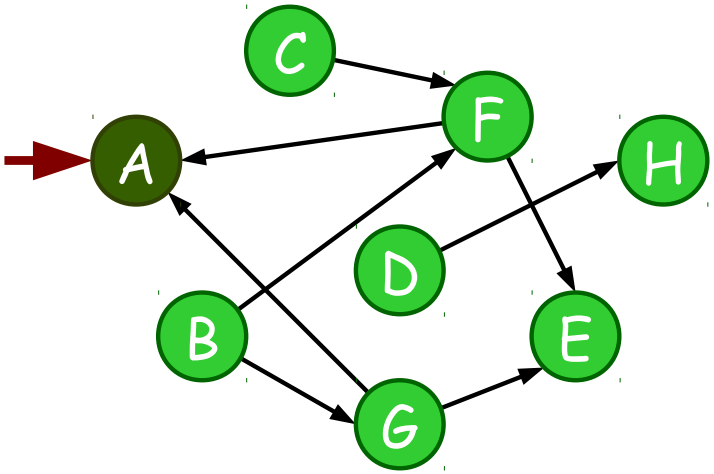


Ordenação topológica: exemplo



A

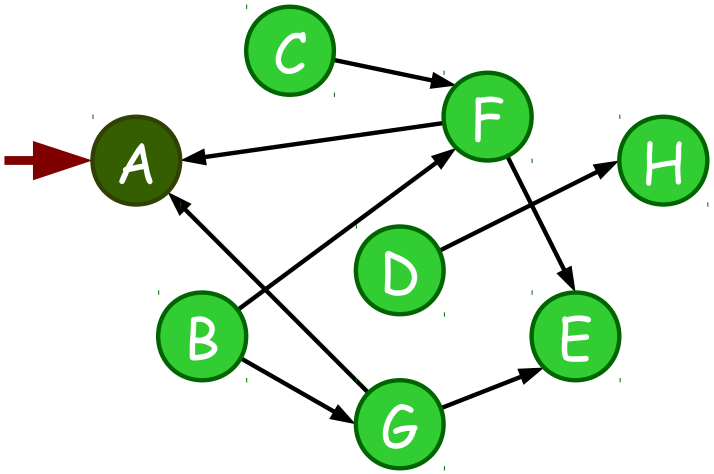
Ordenação topológica: exemplo



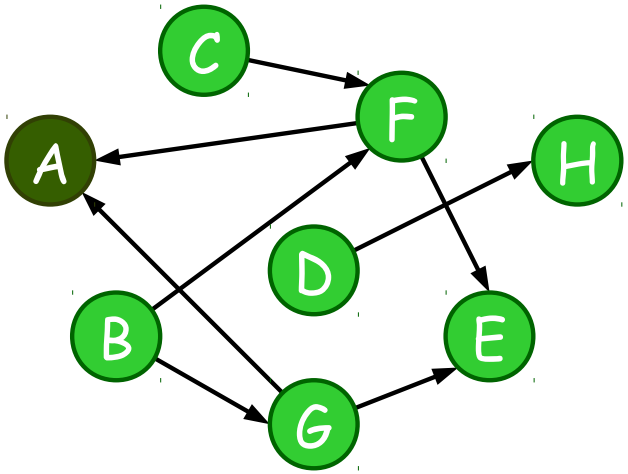
A

pular

Ordenação topológica: exemplo

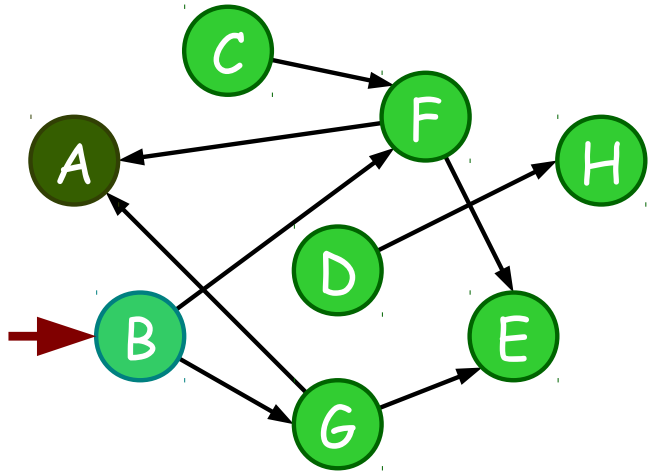


Ordenação topológica: exemplo



pular

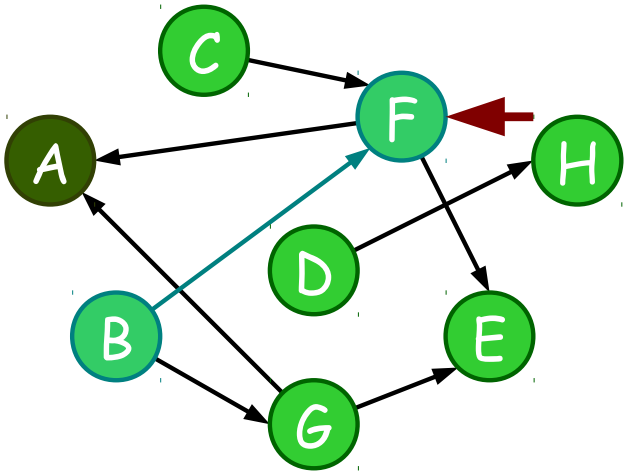
Ordenação topológica: exemplo

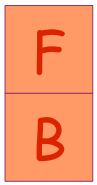


B

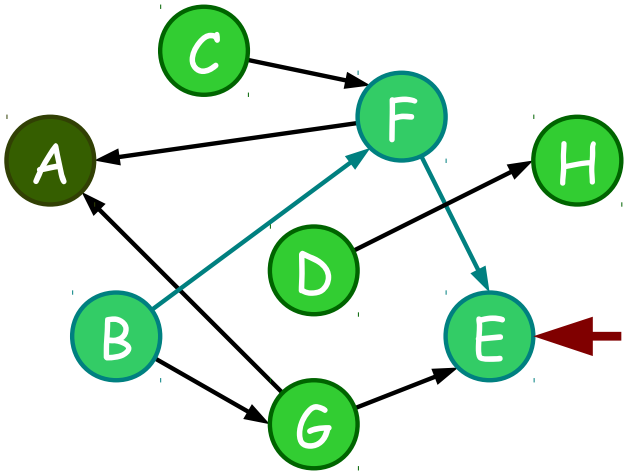
A

Ordenação topológica: exemplo



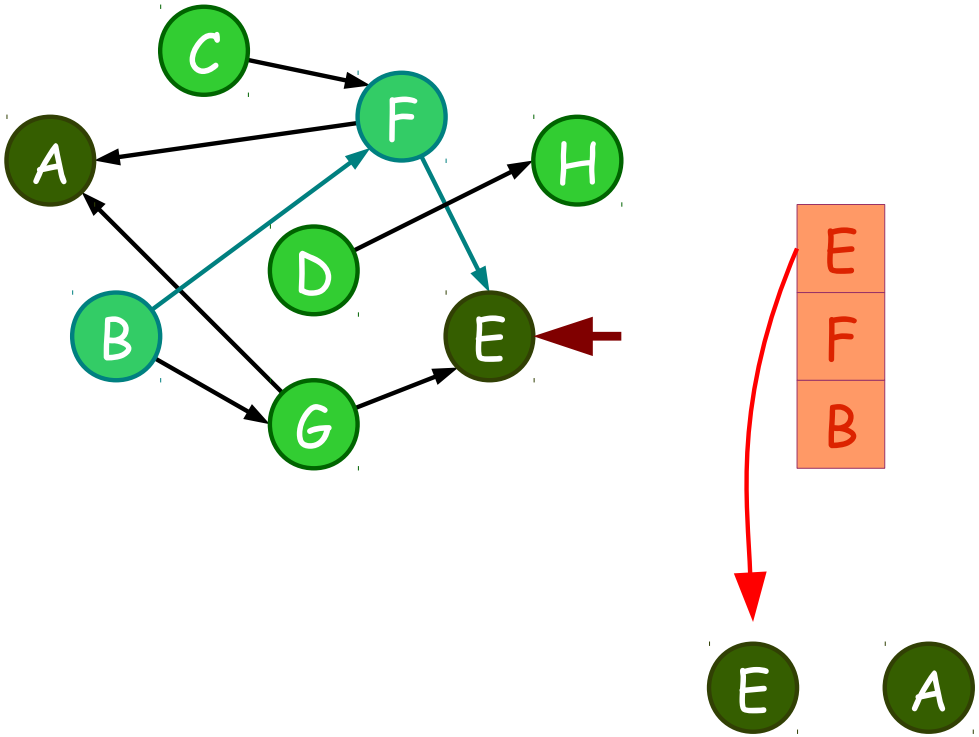


Ordenação topológica: exemplo

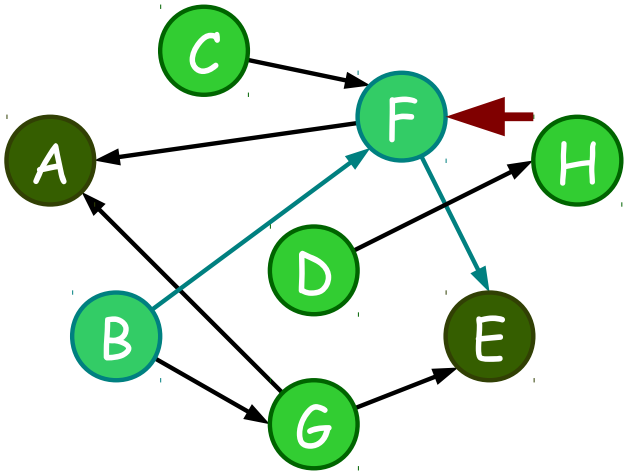


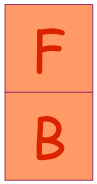
pular

Ordenação topológica: exemplo

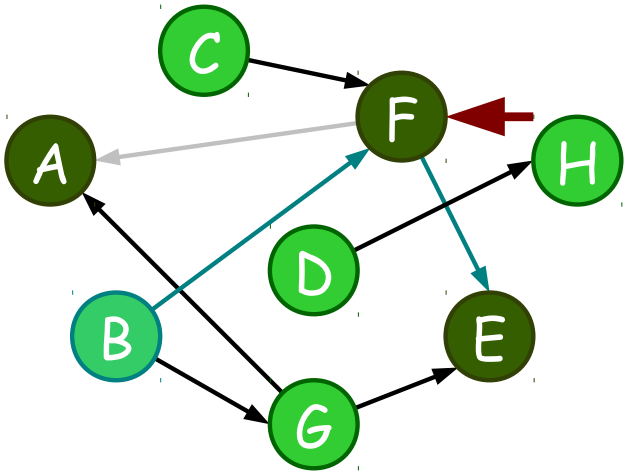


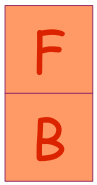
Ordenação topológica: exemplo



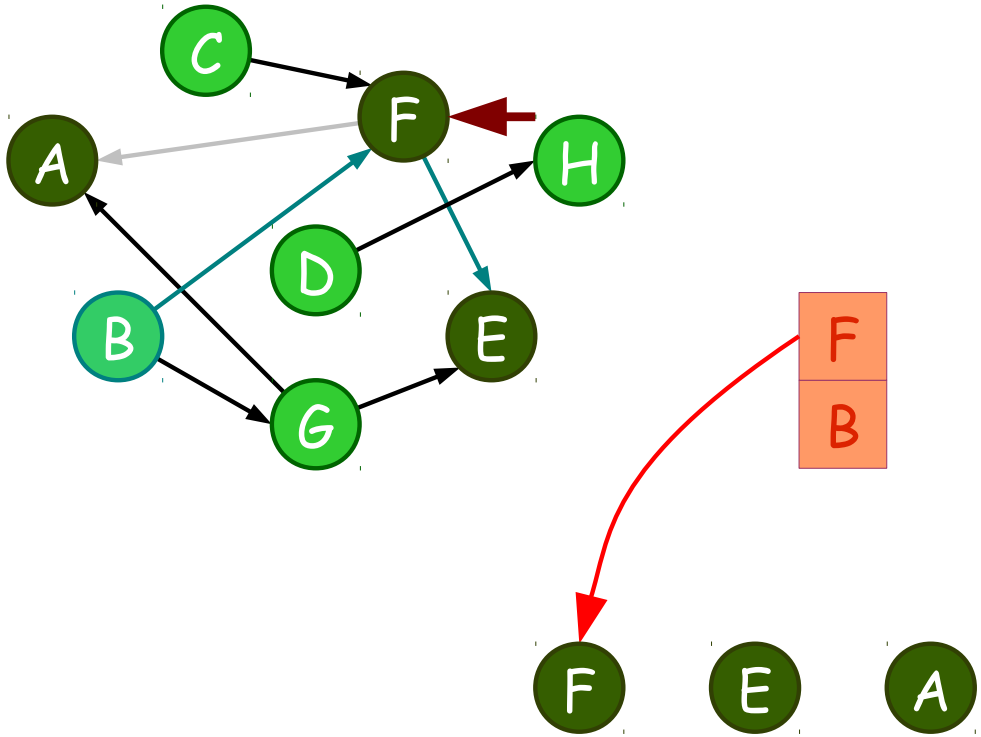


Ordenação topológica: exemplo

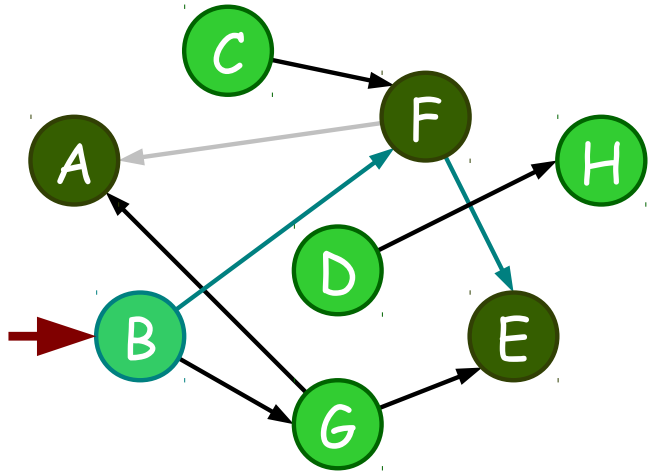




Ordenação topológica: exemplo

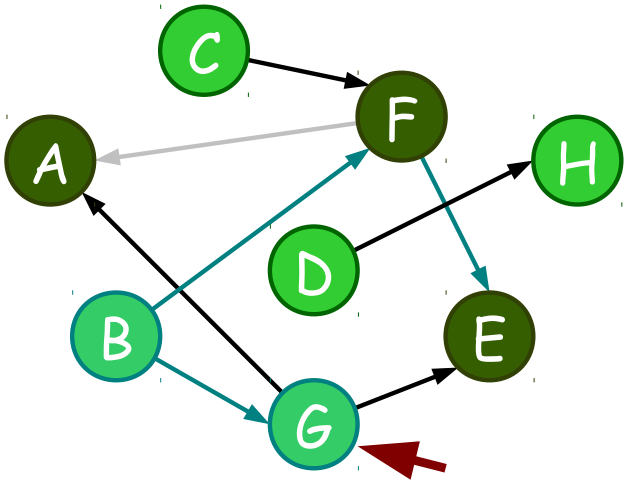


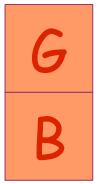
Ordenação topológica: exemplo



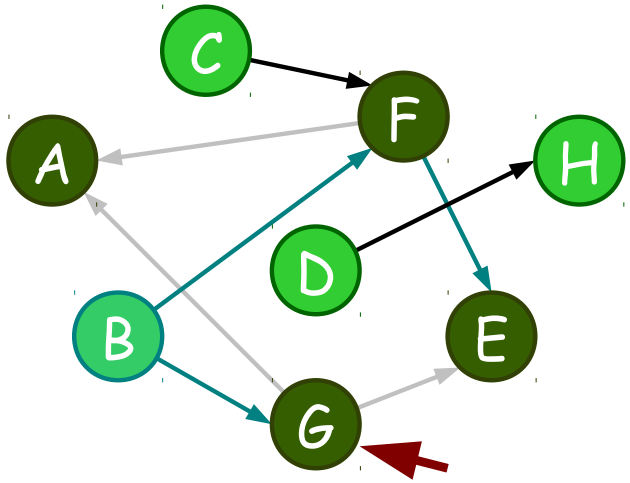
pular

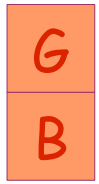
Ordenação topológica: exemplo





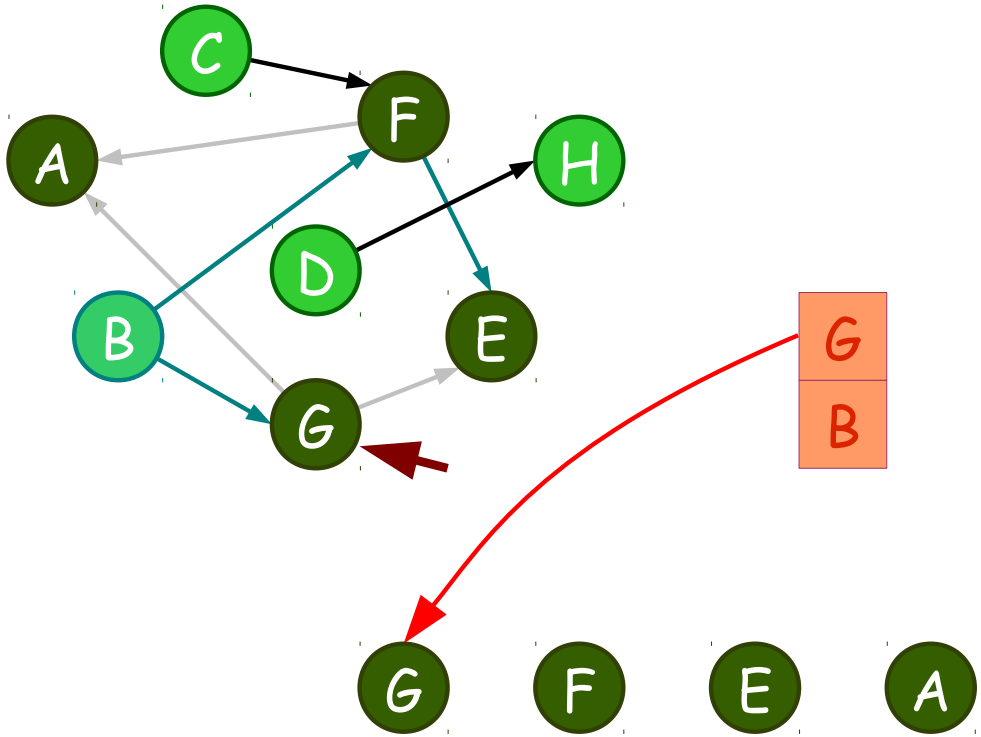
Ordenação topológica: exemplo



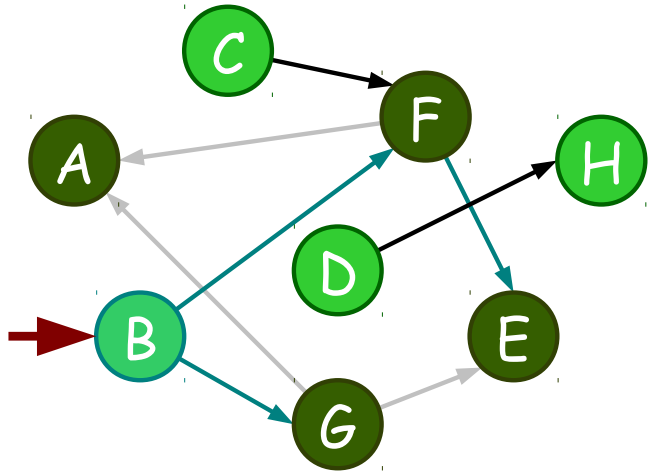


pular

Ordenação topológica: exemplo

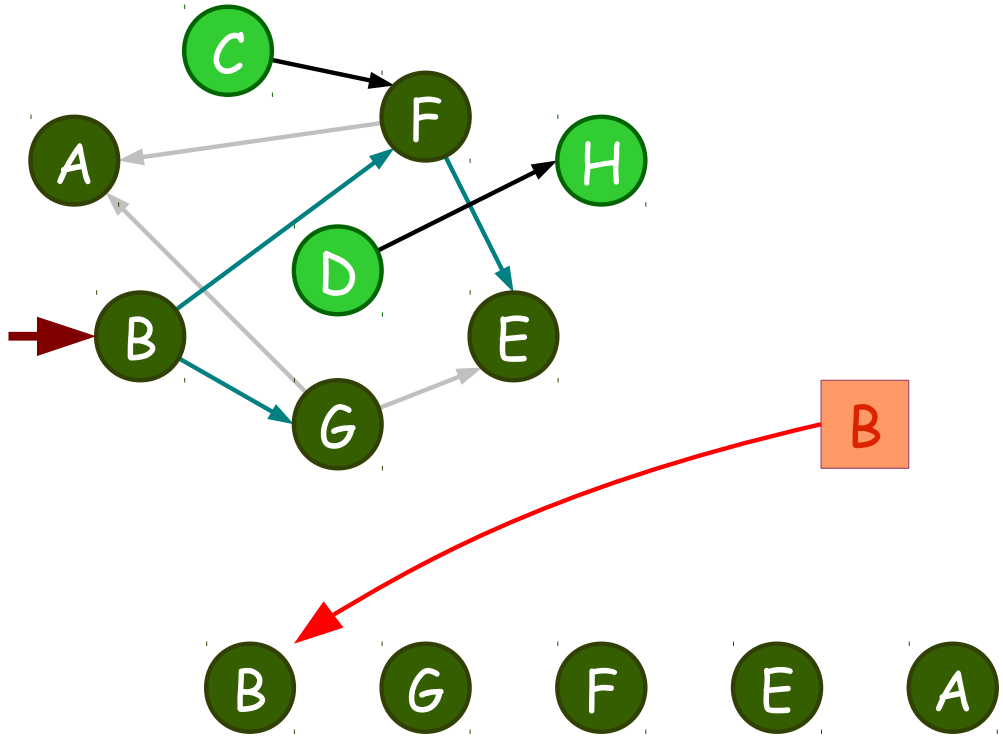


Ordenação topológica: exemplo

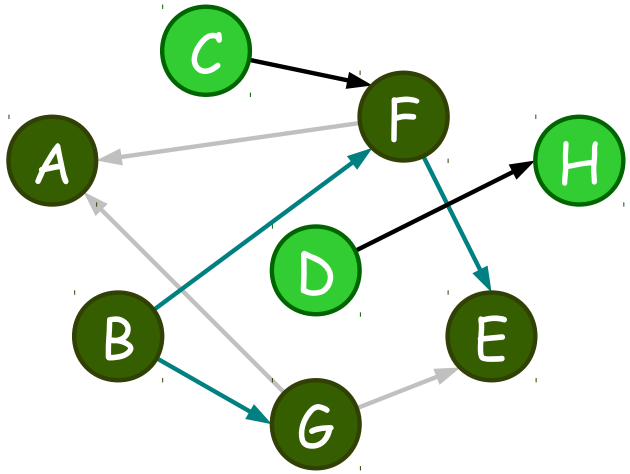


pular

Ordenação topológica: exemplo

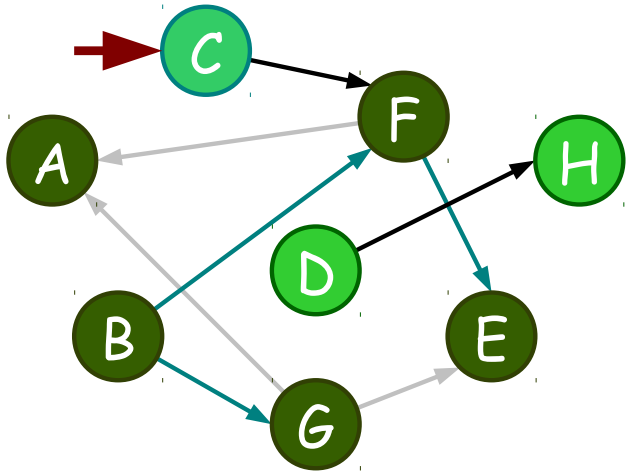


Ordenação topológica: exemplo



pular

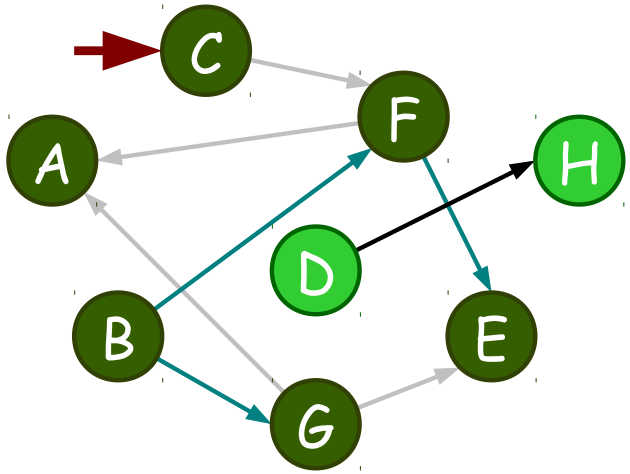
Ordenação topológica: exemplo



C

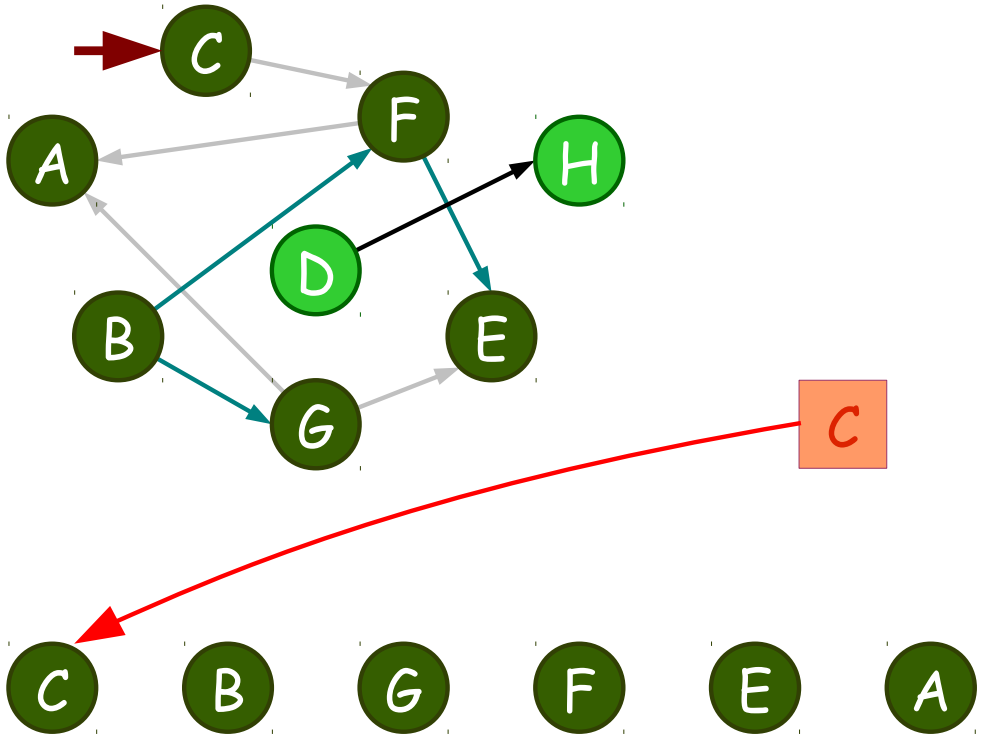
B G F E A

Ordenação topológica: exemplo

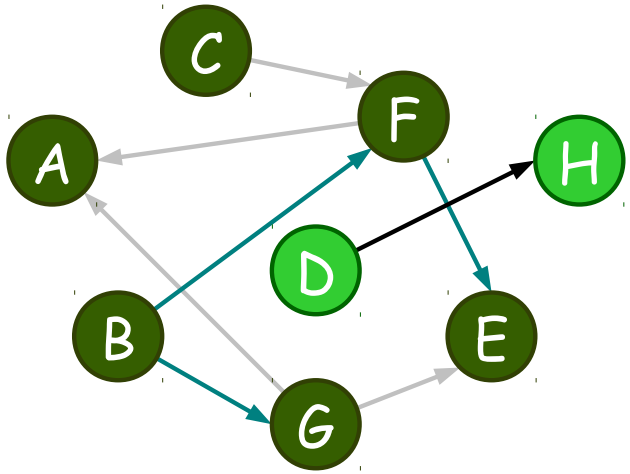


pular

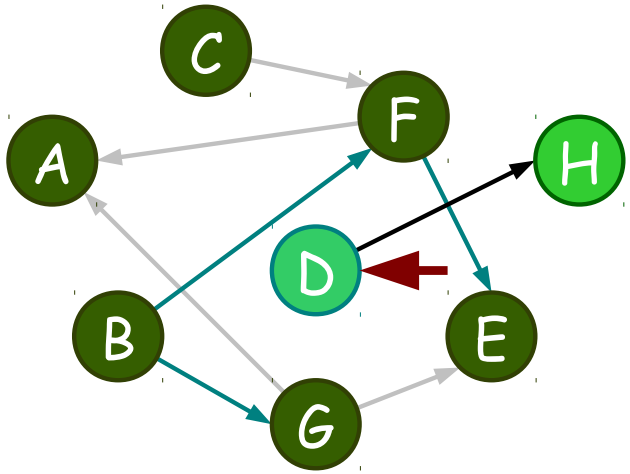
Ordenação topológica: exemplo



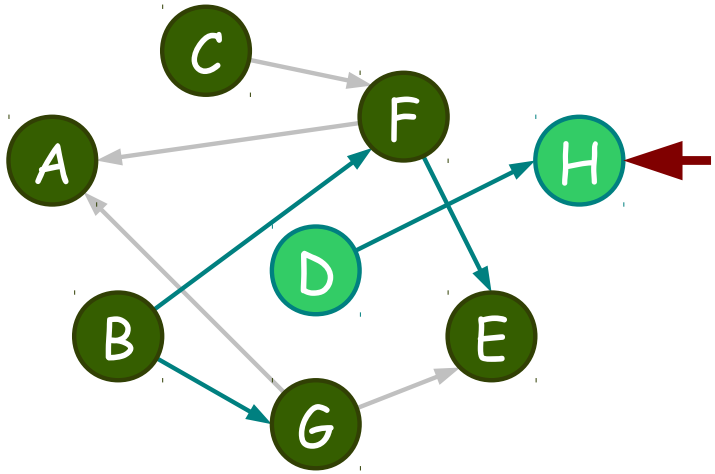
Ordenação topológica: exemplo

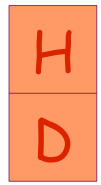


Ordenação topológica: exemplo

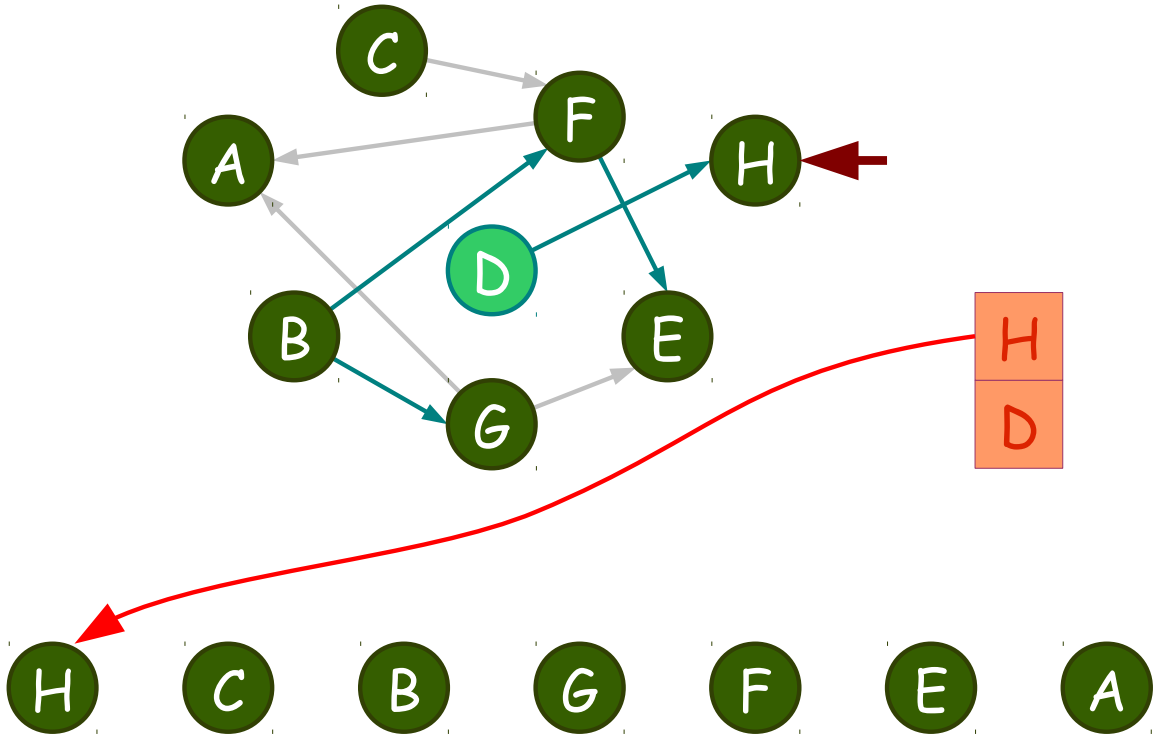


Ordenação topológica: exemplo

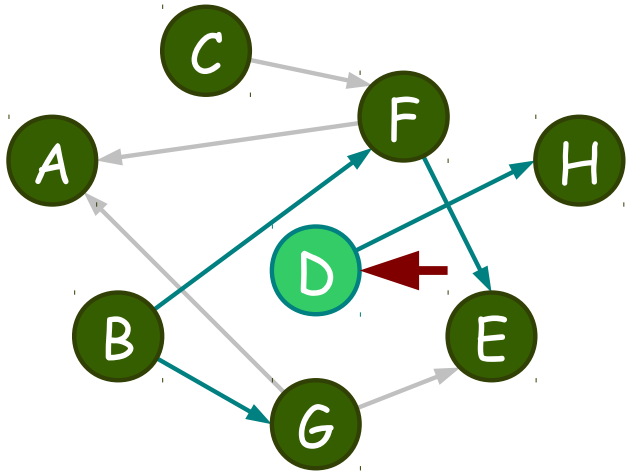




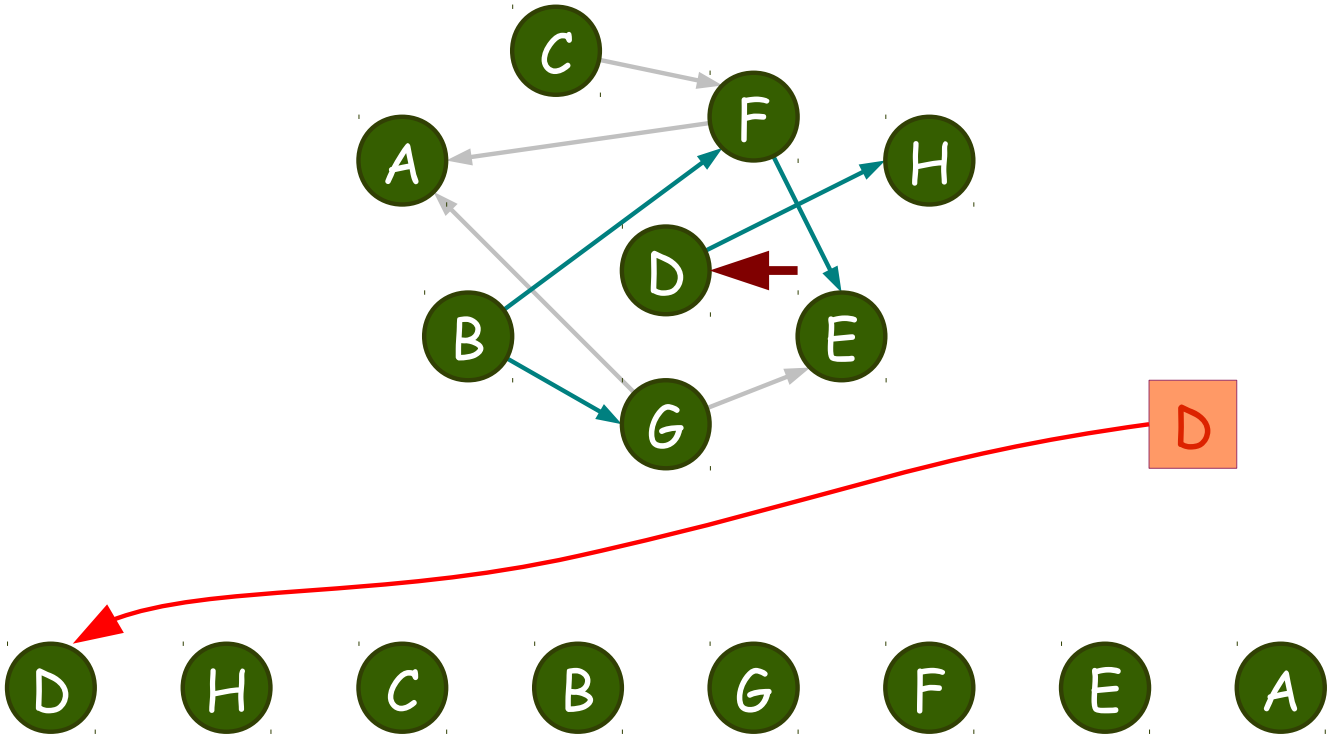
Ordenação topológica: exemplo



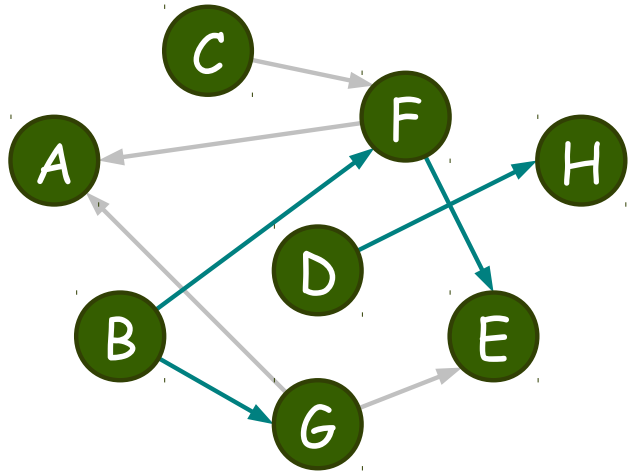
Ordenação topológica: exemplo



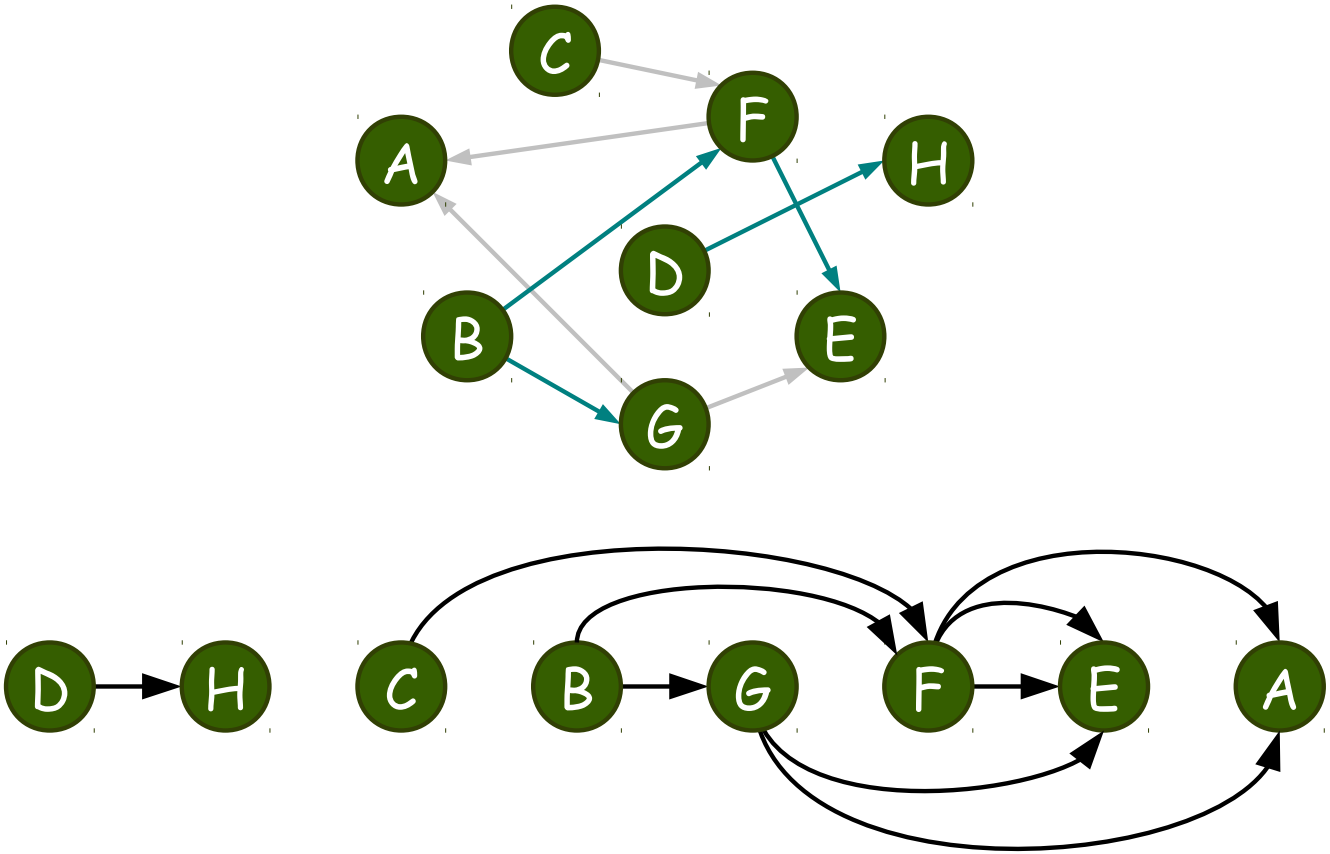
Ordenação topológica: exemplo



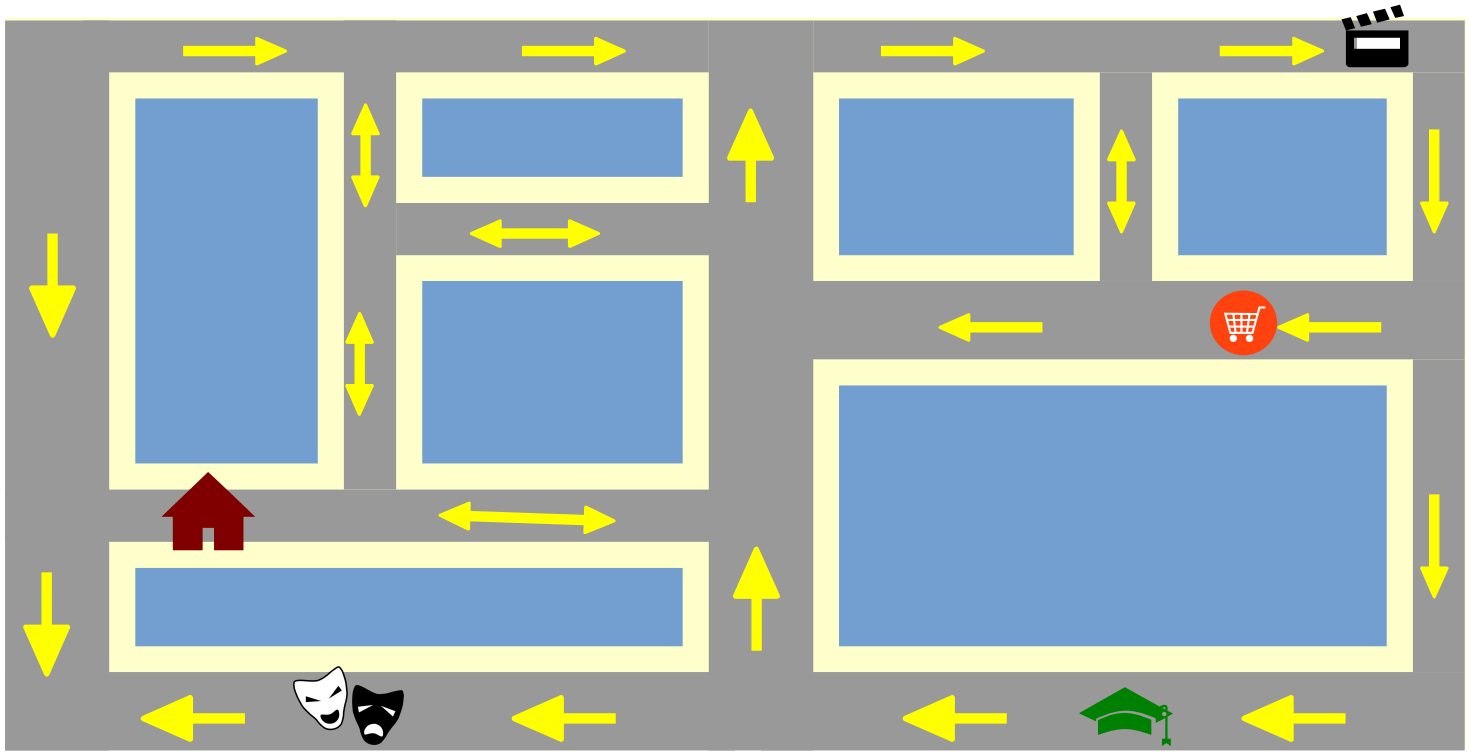
Ordenação topológica: exemplo



Ordenação topológica: exemplo



Mais um problema



Problema

Você vai rotineiramente da sua casa até vários locais (supermercado, teatro, cinema, universidade etc.). Qual o menor caminho da sua casa até cada um deles?

Problema do caminho mínimo

Árvore de caminhos mínimos

A árvore de caminhos mínimos a partir de um vértice de origem s em G é uma árvore, subgrafo de G, tal que, para todo vértice v:

- o caminho entre s e v na árvore é um menor caminho do grafo
- a distância entre s e v está associada ao nó v.

Problema do caminho mínimo

Árvore de caminhos mínimos

A árvore de caminhos mínimos a partir de um vértice de origem s em G é uma árvore, subgrafo de G, tal que, para todo vértice v:

- o caminho entre s e v na árvore é um menor caminho do grafo
- a distância entre s e v está associada ao nó v.

Aplicações

- caminhos em mapas
- navegação
- planejamento de tráfego
- roteamento de pacotes de rede
- agendamento de tarefas
- etc...

Problema do caminho mínimo

Árvore de caminhos mínimos

A árvore de caminhos mínimos a partir de um vértice de origem s em G é uma árvore, subgrafo de G, tal que, para todo vértice v:

- o caminho entre s e v na árvore é um menor caminho do grafo
- a distância entre s e v está associada ao nó v.

Aplicações

- caminhos em mapas
- navegação
- planejamento de tráfego
- roteamento de pacotes de rede
- agendamento de tarefas
- etc...

Vários algoritmos: Dijkstra, Bellman–Ford etc.

Problema do caminho mínimo

Árvore de caminhos mínimos

A árvore de caminhos mínimos a partir de um vértice de origem s em G é uma árvore, subgrafo de G, tal que, para todo vértice v:

- o caminho entre s e v na árvore é um menor caminho do grafo
- a distância entre s e v está associada ao nó v.

Aplicações

- caminhos em mapas
- navegação
- planejamento de tráfego
- roteamento de pacotes de rede
- agendamento de tarefas
- etc...

Vários algoritmos: Dijkstra, Bellman–Ford etc.

Algoritmo de Dijkstra

Características

- proposto pelo holandês Edsger Dijkstra (1956)
- "análogo" à busca em largura
- presume pesos não negativos

Algoritmo de Dijkstra

Características

- proposto pelo holandês Edsger Dijkstra (1956)
- "análogo" à busca em largura
- presume pesos não negativos

DIJKSTRA(G, s, w)

1 **for** each vertex $u \in V$
2 $\quad d[u] = \infty$
3 $\quad \pi[u] = \text{NULL}$
4 $d[s] = 0$

Algoritmo de Dijkstra

Características

- proposto pelo holandês Edsger Dijkstra (1956)
- "análogo" à busca em largura
- presume pesos não negativos

DIJKSTRA(G, s, w)

1 **for** each vertex $u \in V$
2 $\quad d[u] = \infty$
3 $\quad \pi[u] = \text{NULL}$
4 $d[s] = 0$
5 $Q = \text{CREATE-MINHEAP}(V, d)$

Algoritmo de Dijkstra

Características

- proposto pelo holandês Edsger Dijkstra (1956)
- "análogo" à busca em largura
- presume pesos não negativos

DIJKSTRA(G, s, w)

1 **for** each vertex $u \in V$
2 $\quad d[u] = \infty$
3 $\quad \pi[u] = \text{NULL}$
4 $d[s] = 0$
5 $Q = \text{CREATE-MINHEAP}(V, d)$
6 **while** $Q \neq \emptyset$
7 $\quad u = \text{REMOVE-MIN}(Q)$
8 $\quad$ **for** each vertex $v \in Adj(u)$

Algoritmo de Dijkstra

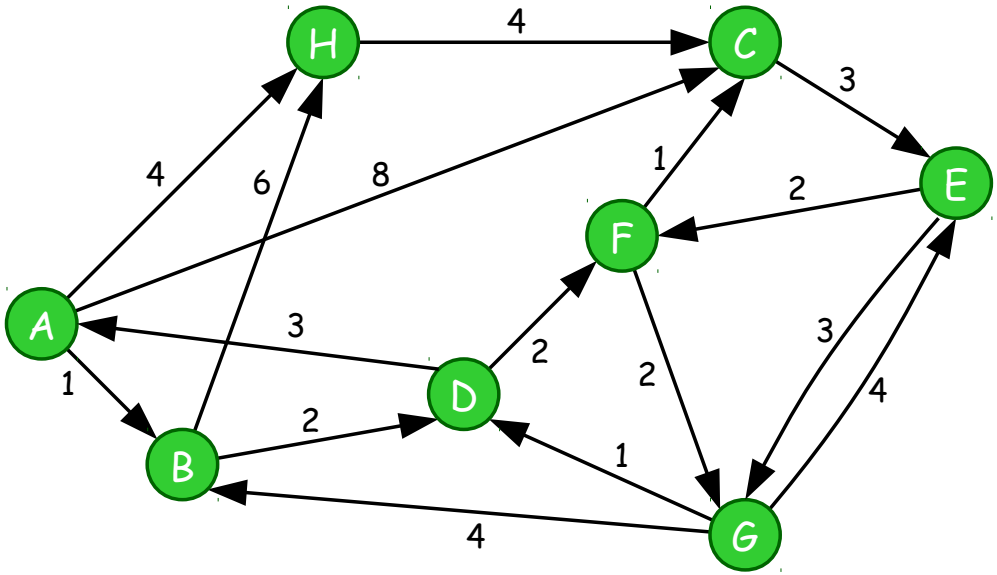
Características

- proposto pelo holandês Edsger Dijkstra (1956)
- "análogo" à busca em largura
- presume pesos não negativos

DIJKSTRA(G, s, w)

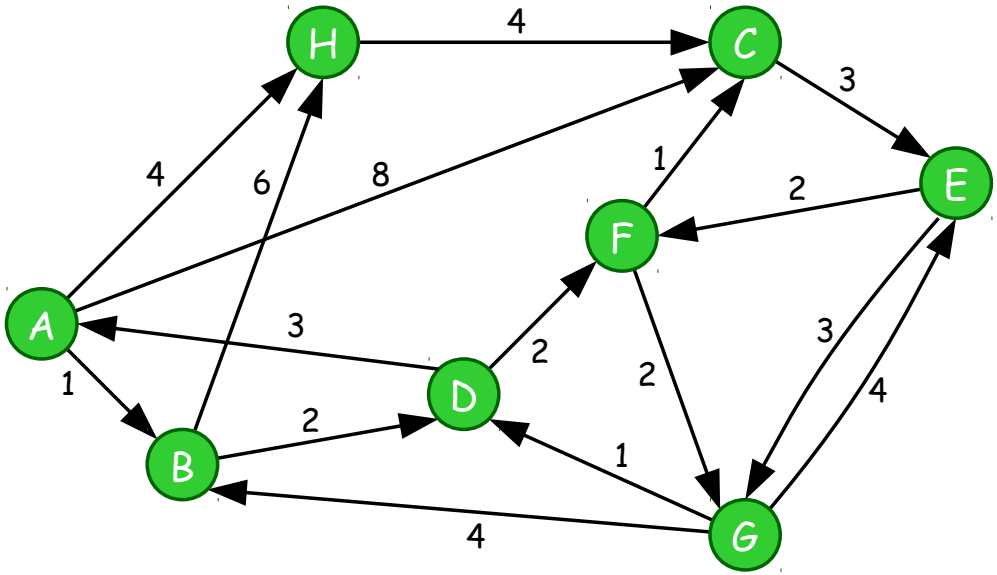
```
1  for each vertex u ∈ V
2      d[u] = ∞
3      π[u] = NULL
4  d[s] = 0
5  Q = CREATE-MINHEAP(V, d)
6  while Q ≠ ∅
7      u = REMOVE-MIN(Q)
8      for each vertex v ∈ Adj(u)
9          if d[v] > d[u] + w[u, v]
10             d[v] = d[u] + w[u, v]
11             π[v] = u
```

Algoritmo de Dijkstra: exemplo



pular

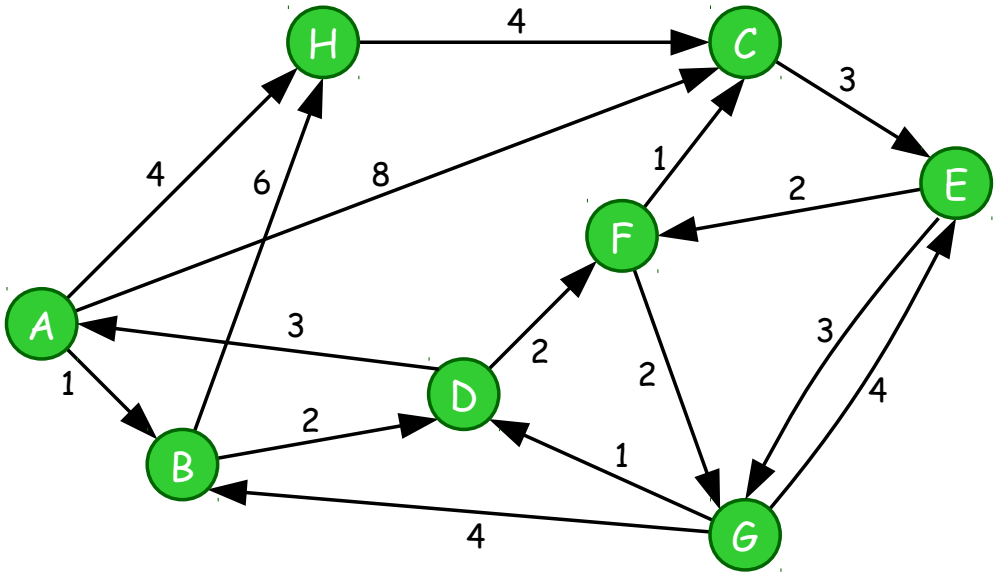
Algoritmo de Dijkstra: exemplo



A	0
B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	∞

pular

Algoritmo de Dijkstra: exemplo



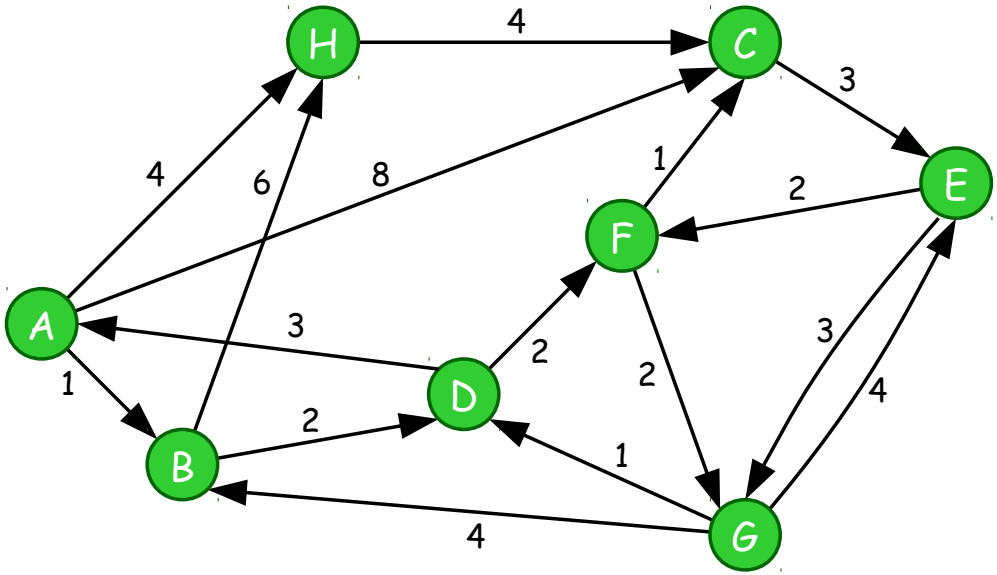
FILA

A	0
B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	∞

mínimo

pular

Algoritmo de Dijkstra: exemplo

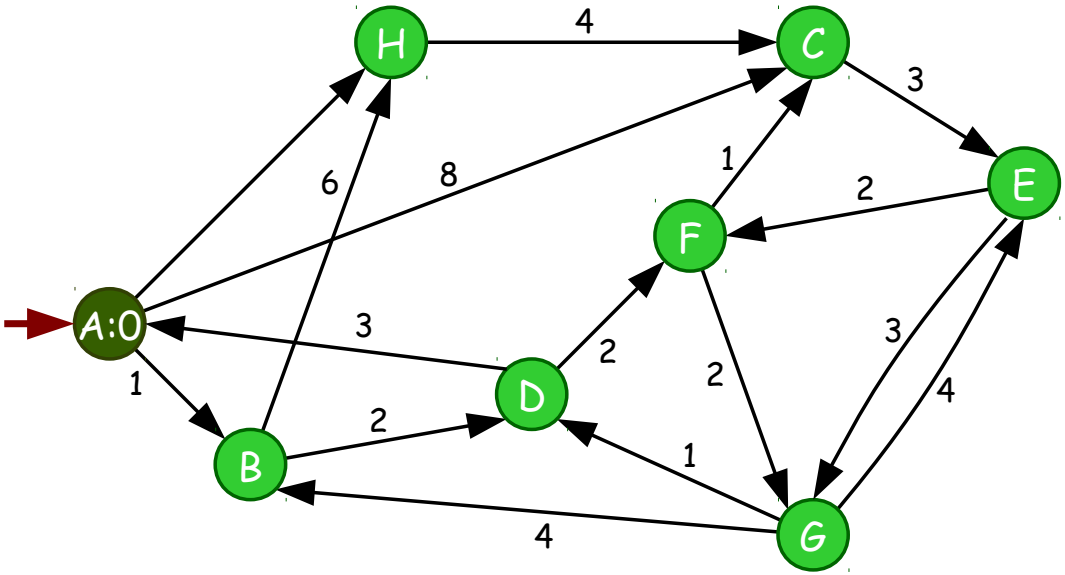


FILA

A	0
B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	∞

pular

Algoritmo de Dijkstra: exemplo

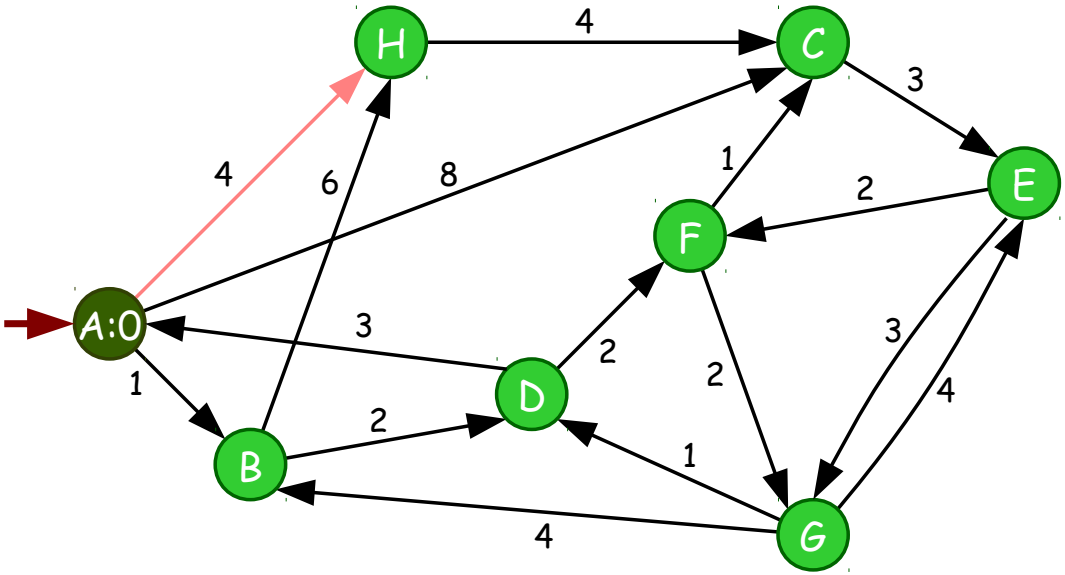


FILA

B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	∞

pular

Algoritmo de Dijkstra: exemplo

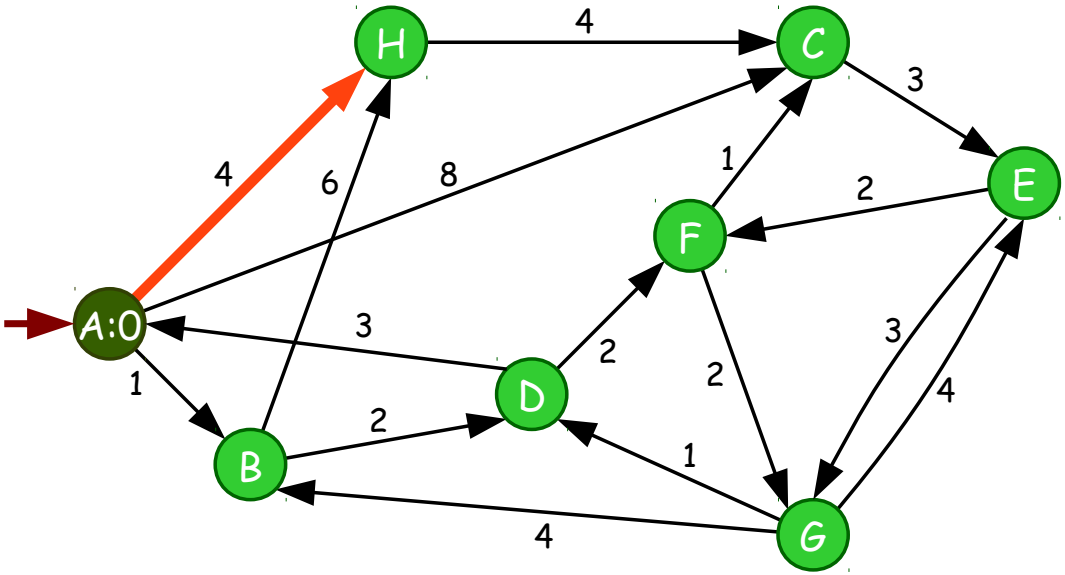


FILA

B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	∞

pular

Algoritmo de Dijkstra: exemplo

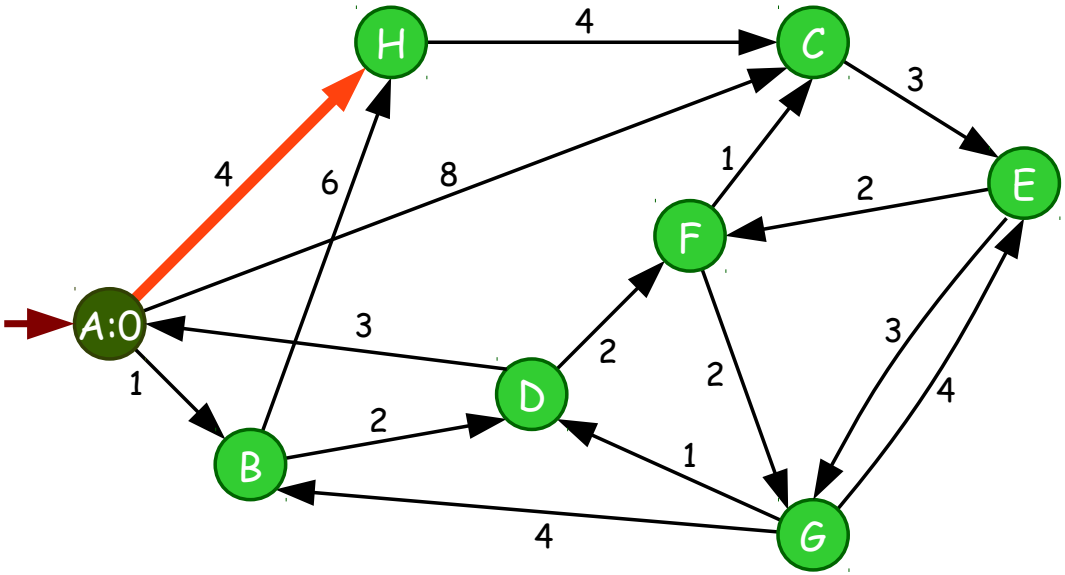


FILA

B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	4

pular

Algoritmo de Dijkstra: exemplo



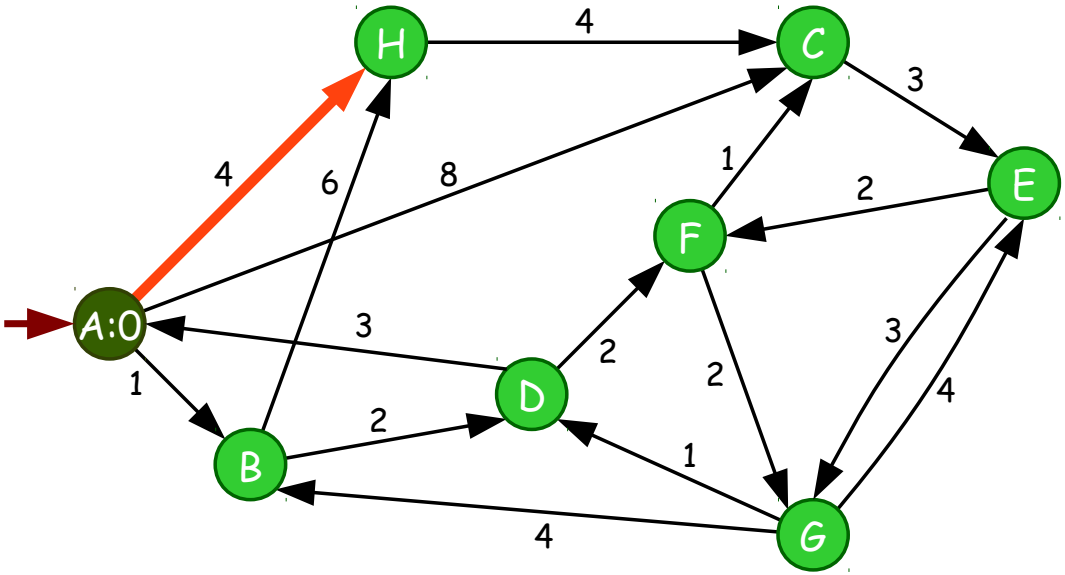
FILA

B	∞
C	∞
D	∞
E	∞
F	∞
G	∞
H	4

atualizamos prioridade

pular

Algoritmo de Dijkstra: exemplo

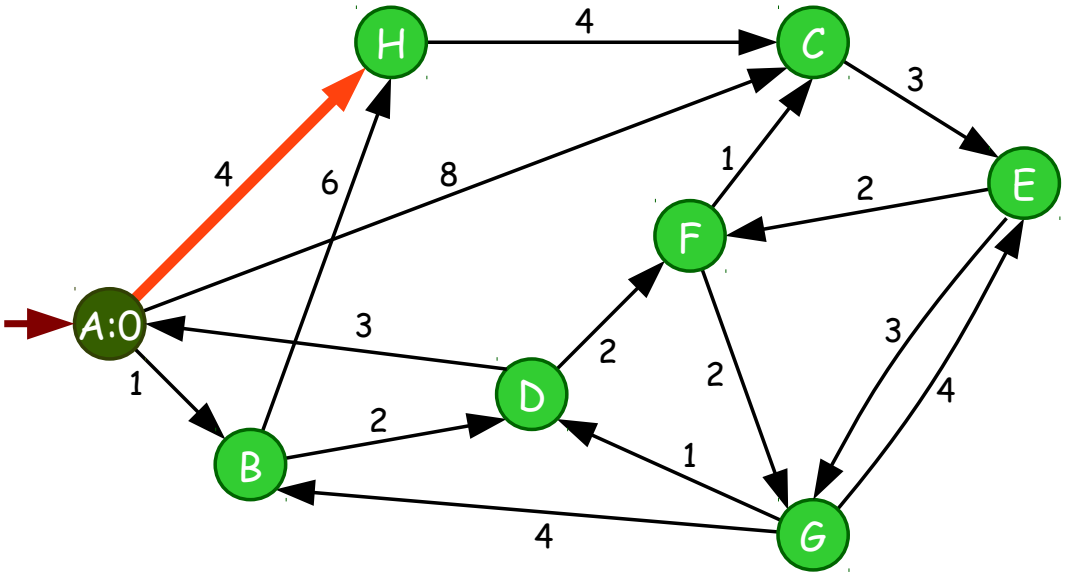


FILA

H	4
B	∞
C	∞
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

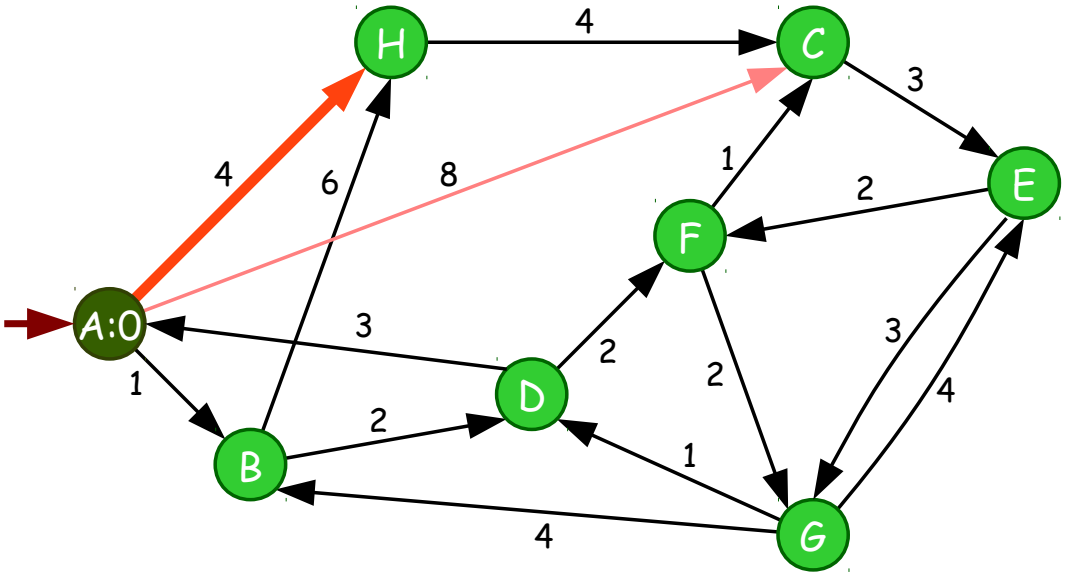


FILA

H	4
B	∞
C	∞
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

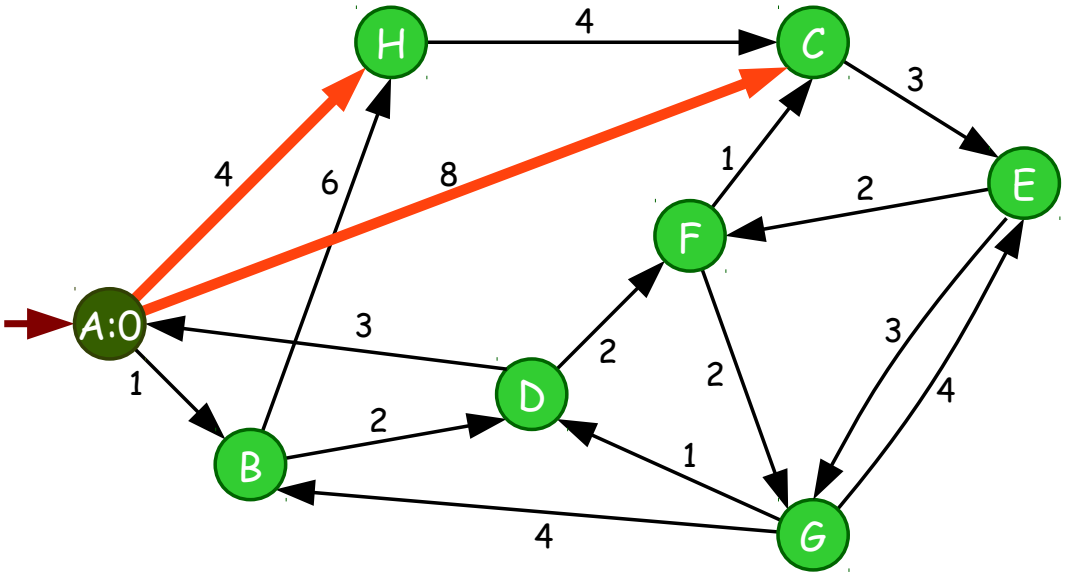


FILA

H	4
B	∞
C	∞
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

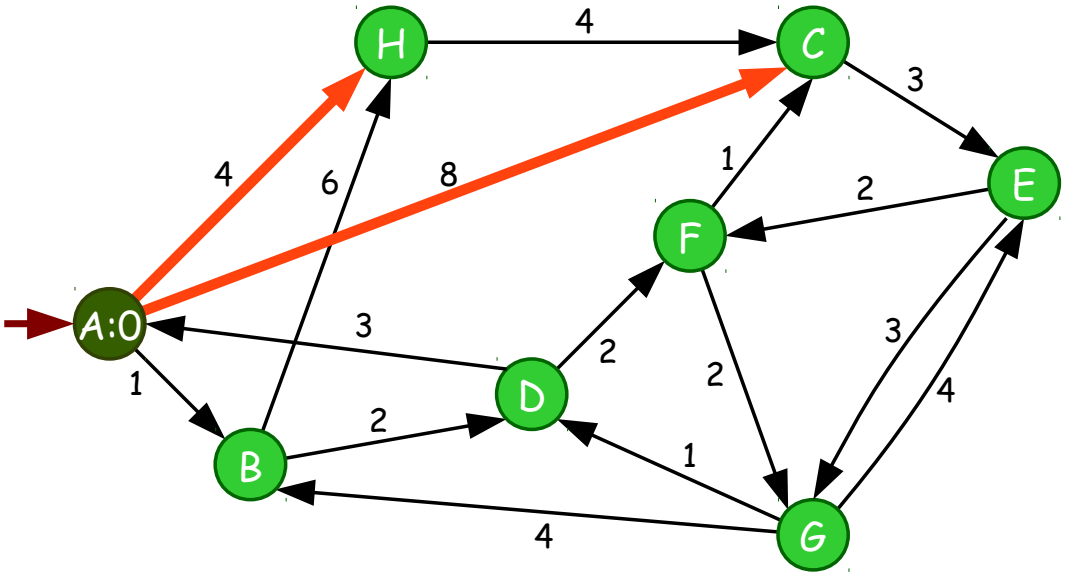


FILA

H	4
B	∞
C	8
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

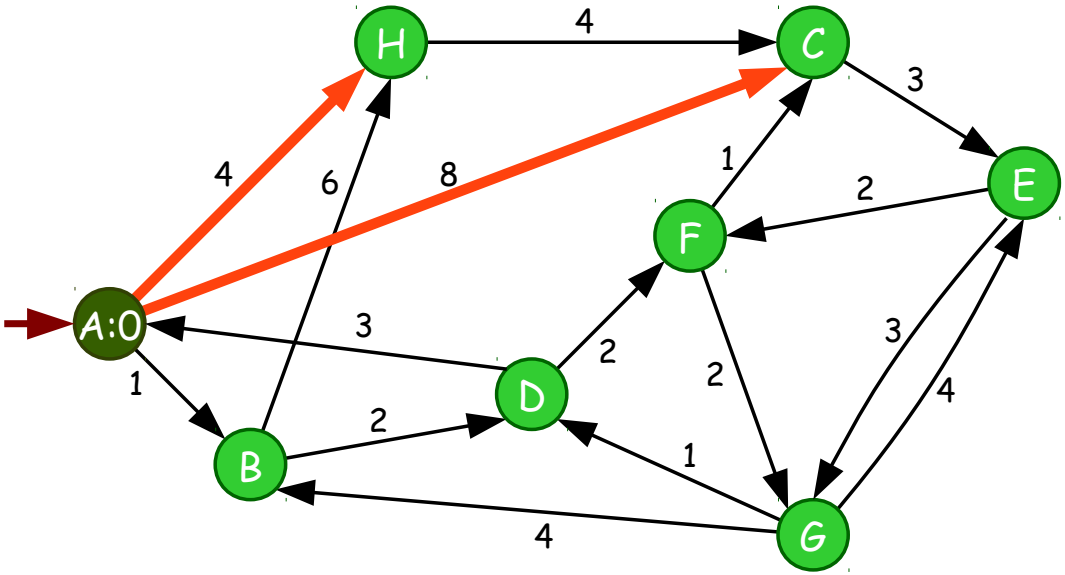


FILA

H	4
C	8
B	∞
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

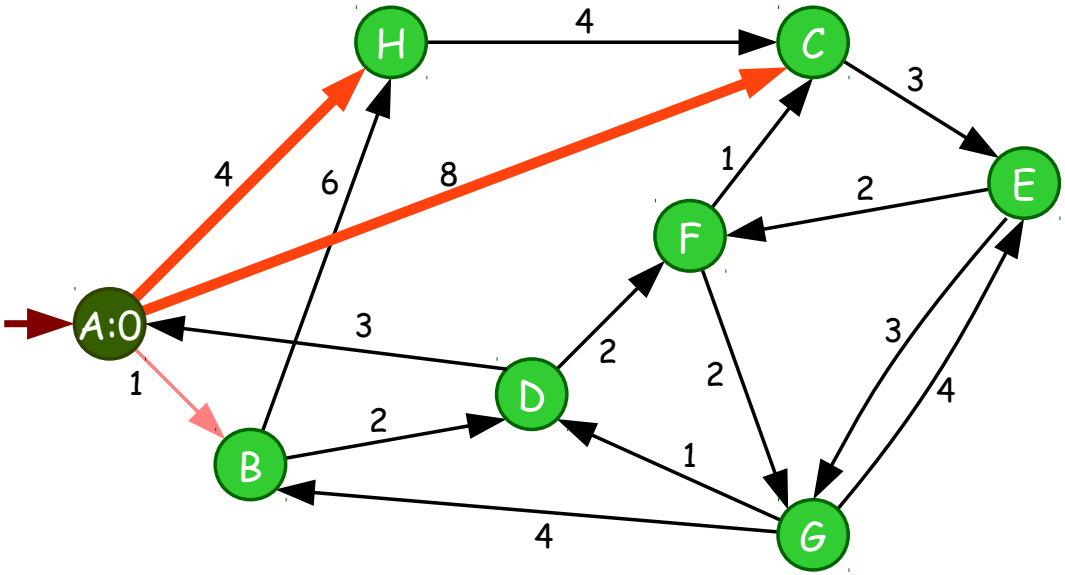


FILA

H	4
C	8
B	∞
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

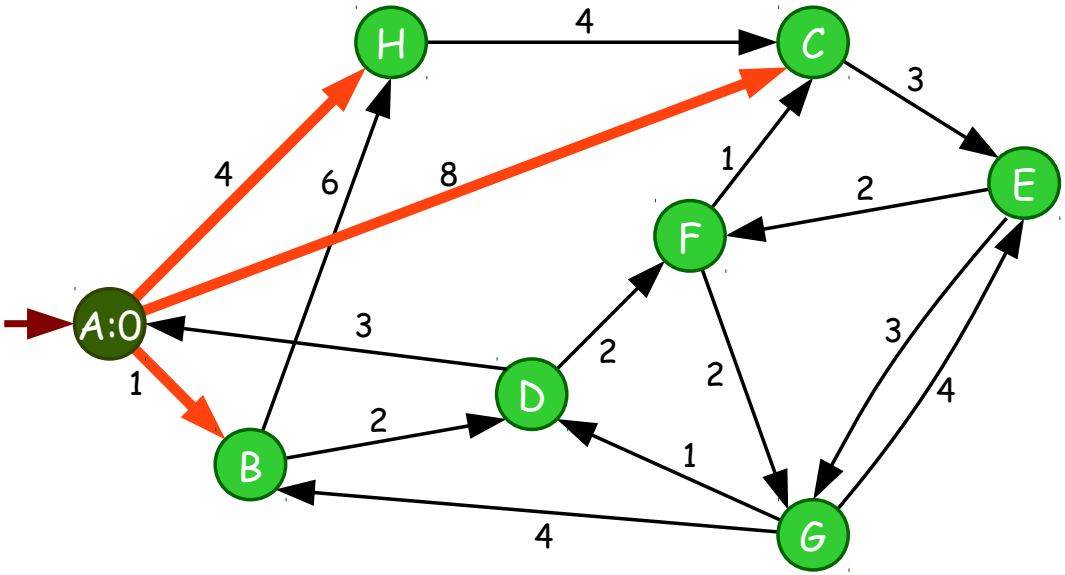


FILA

H	4
C	8
B	∞
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

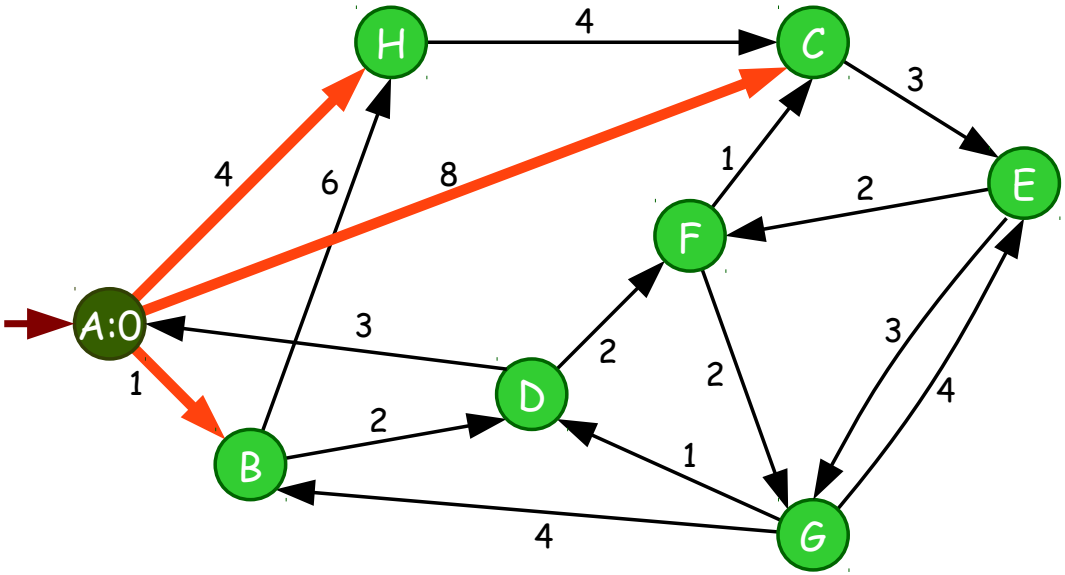


FILA

H	4
C	8
B	1
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

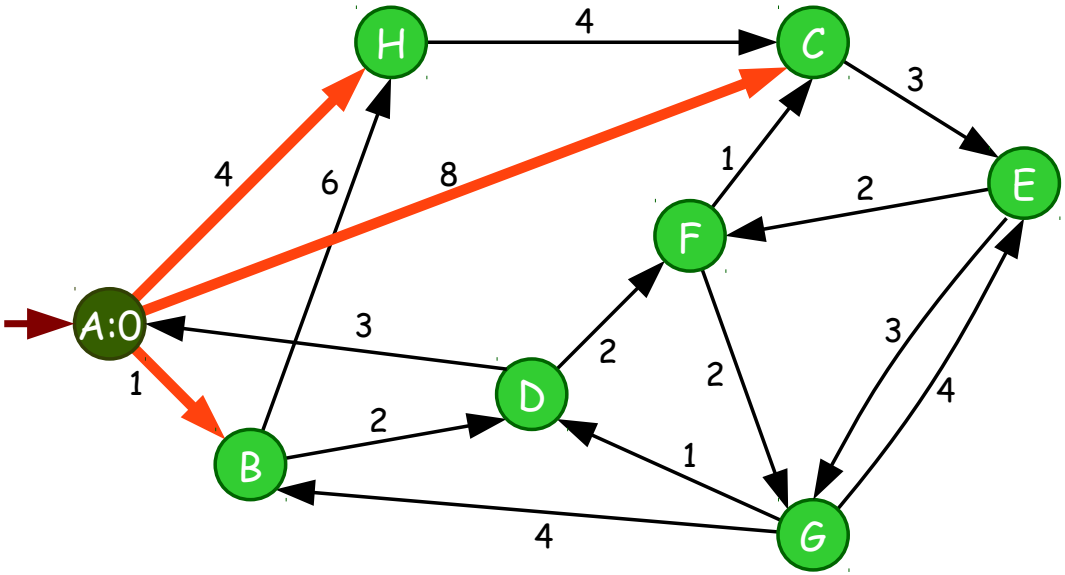


FILA

B	1
H	4
C	8
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

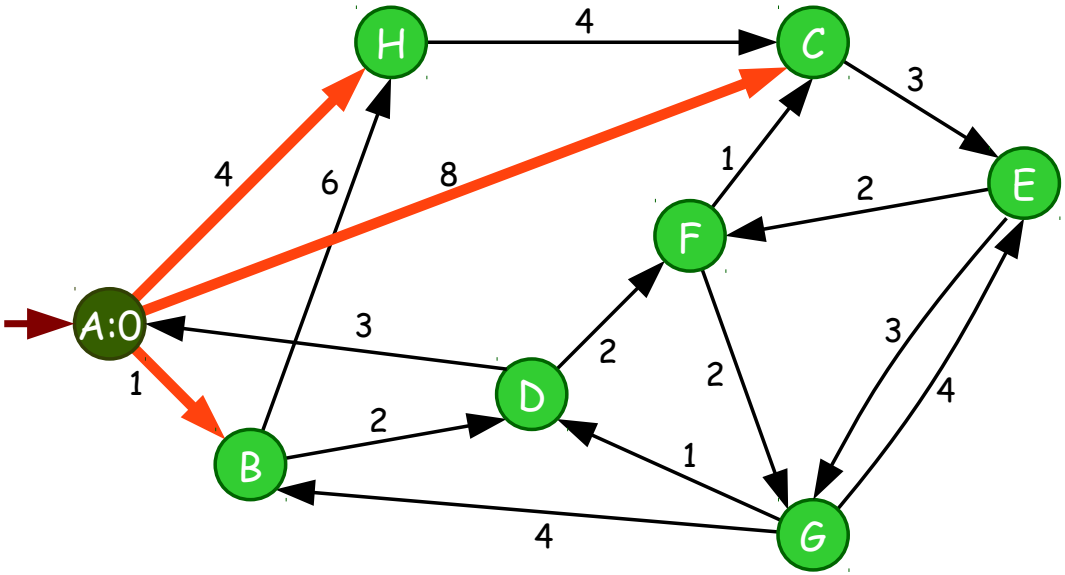


FILA

B	1
H	4
C	8
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

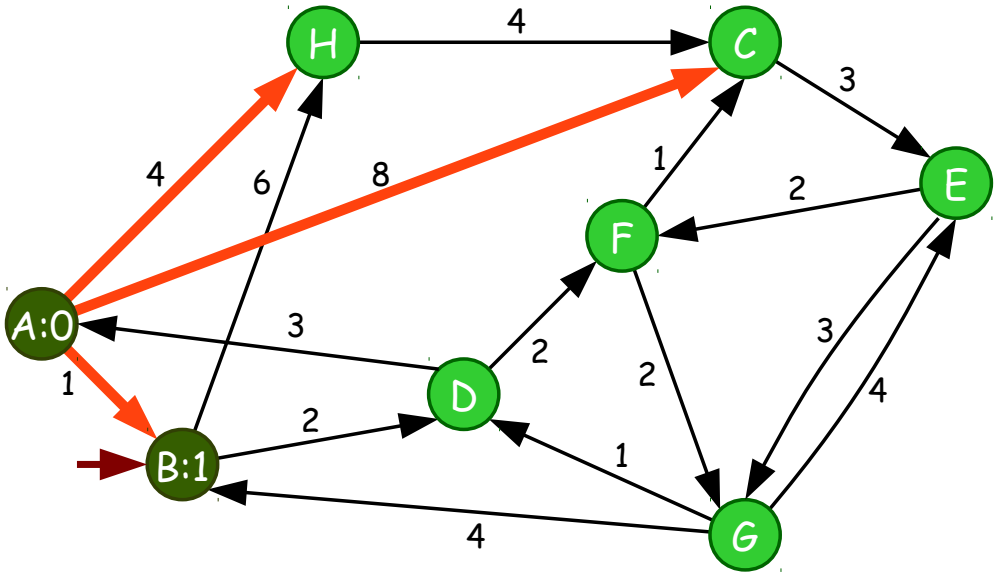


FILA

B	1
H	4
C	8
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

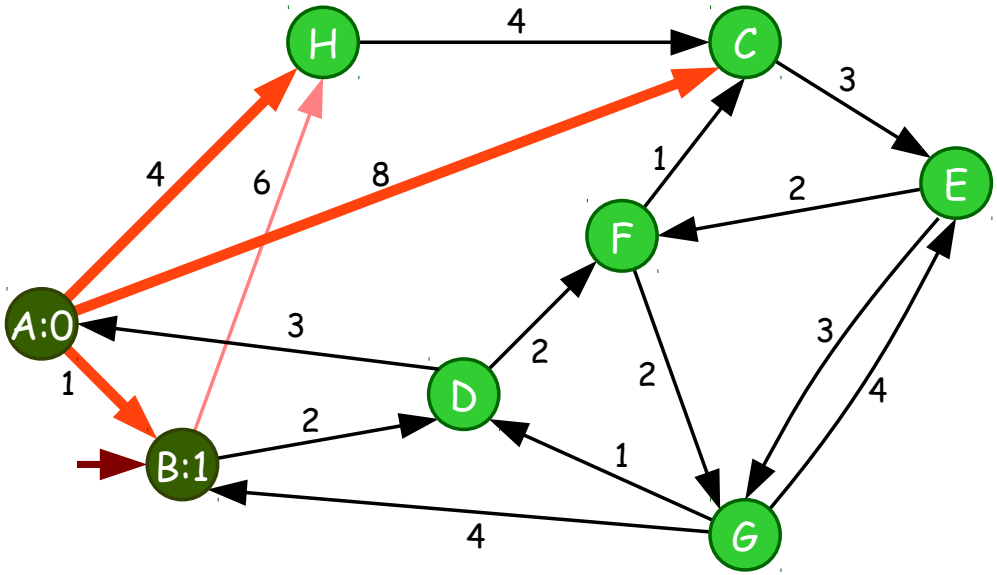


FILA

H	4
C	8
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

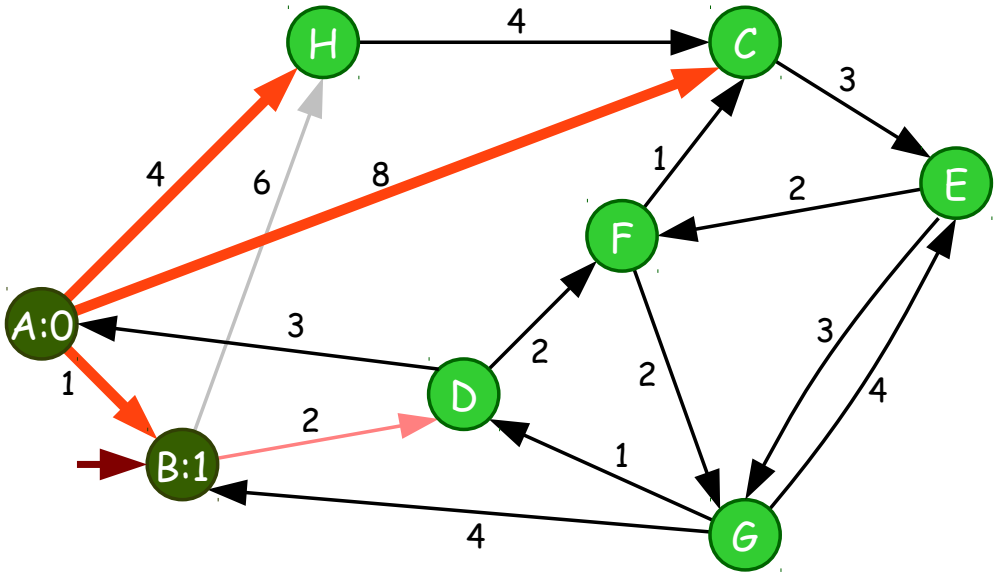


FILA

H	4
C	8
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

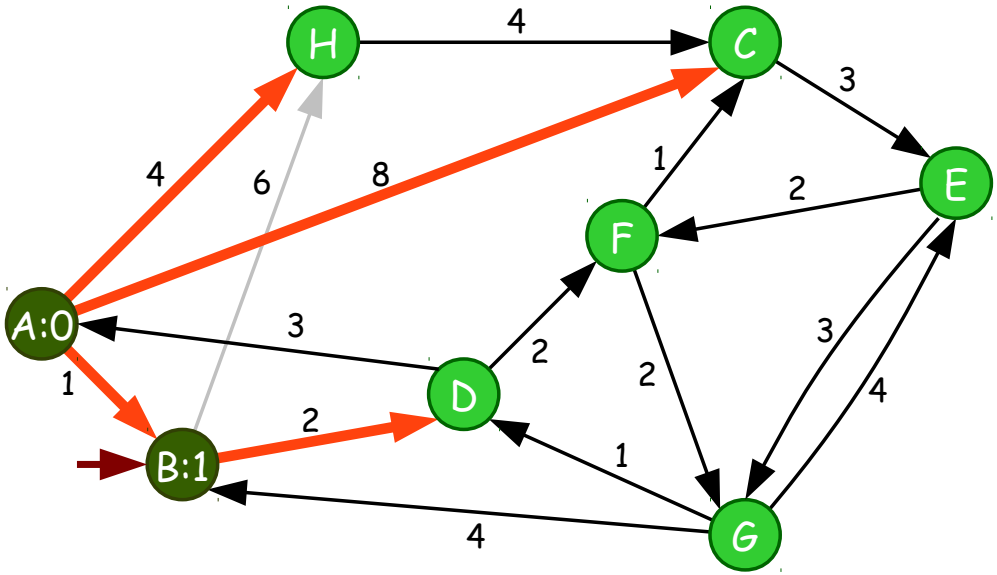


FILA

H	4
C	8
D	∞
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

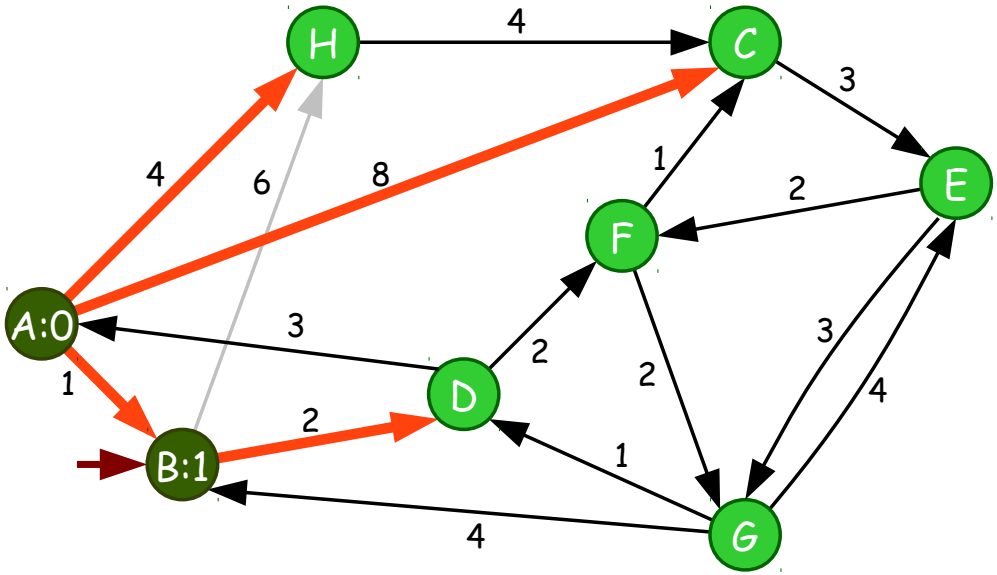


FILA

H	4
C	8
D	3
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

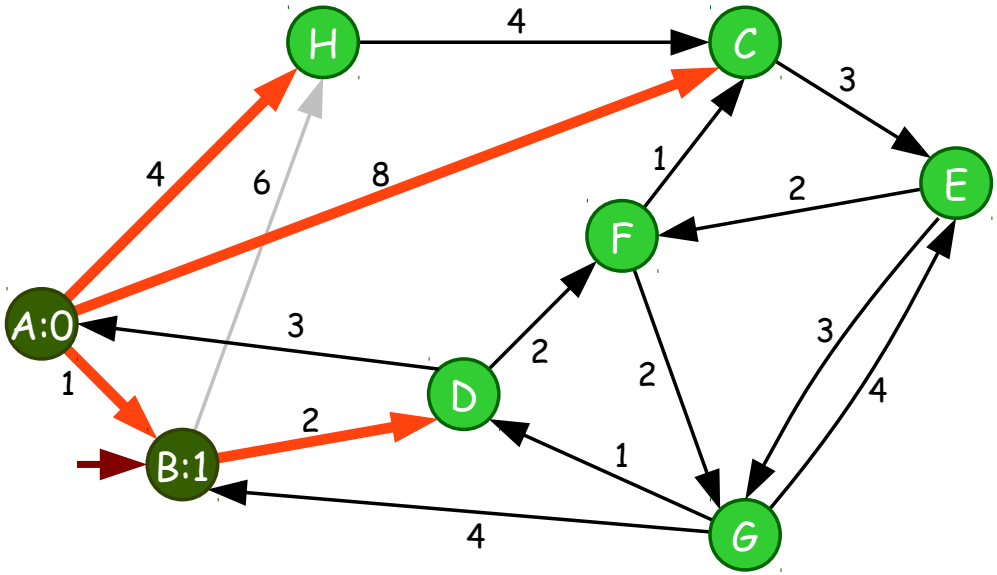


FILA

D	3
H	4
C	8
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

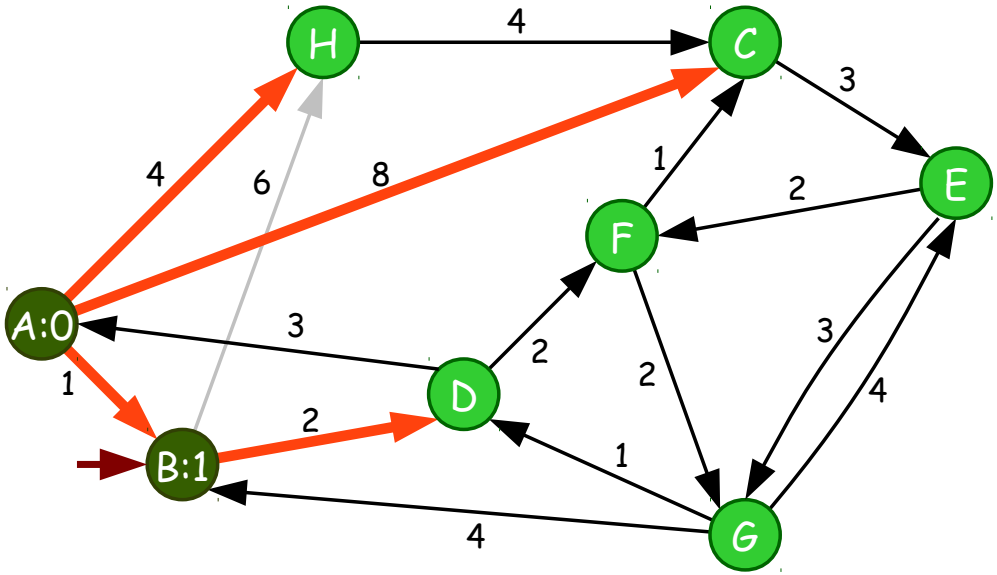


FILA

D	3
H	4
C	8
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

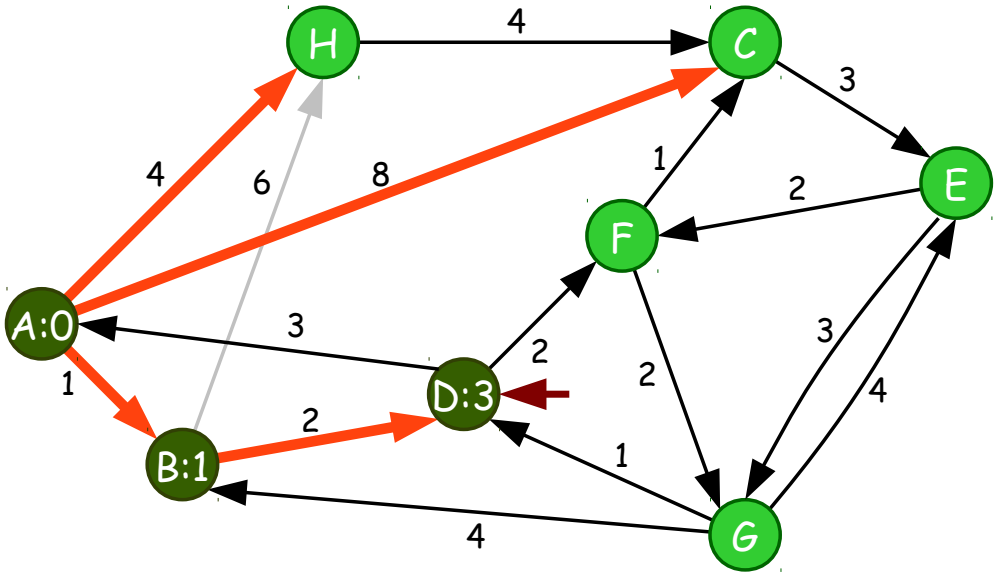


FILA

D	3
H	4
C	8
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

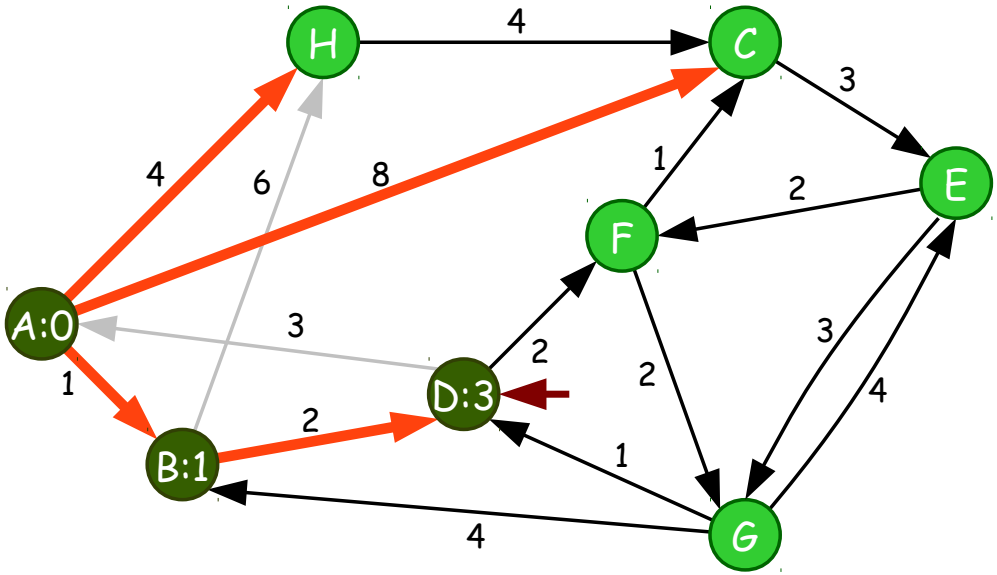


FILA

H	4
C	8
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

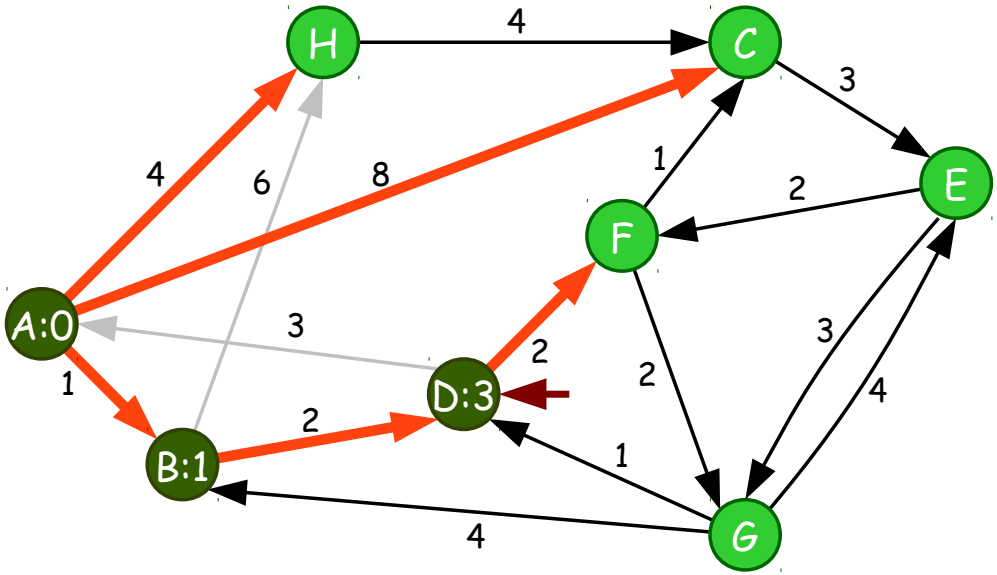


FILA

H	4
C	8
E	∞
F	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

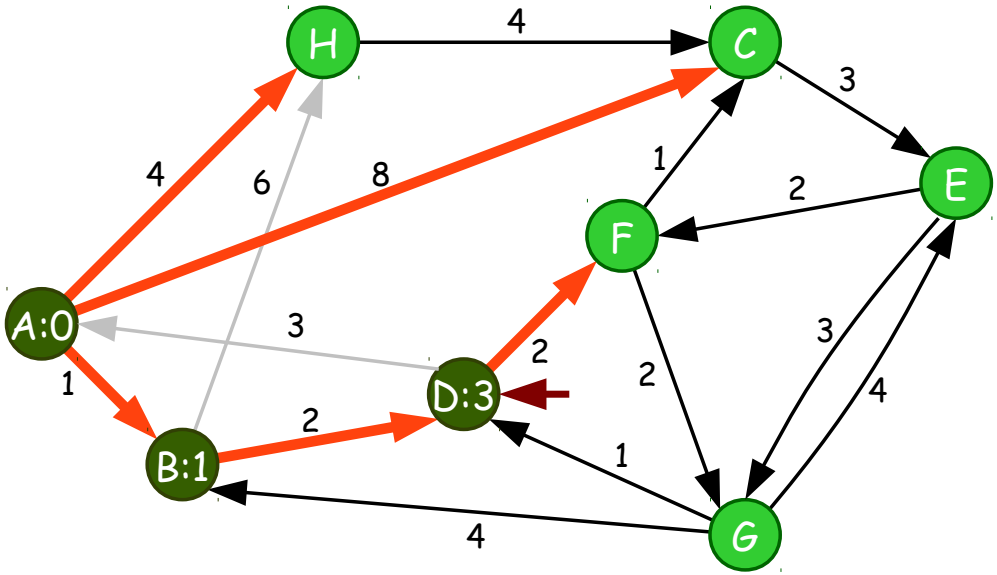


FILA

H	4
C	8
E	∞
F	5
G	∞

pular

Algoritmo de Dijkstra: exemplo

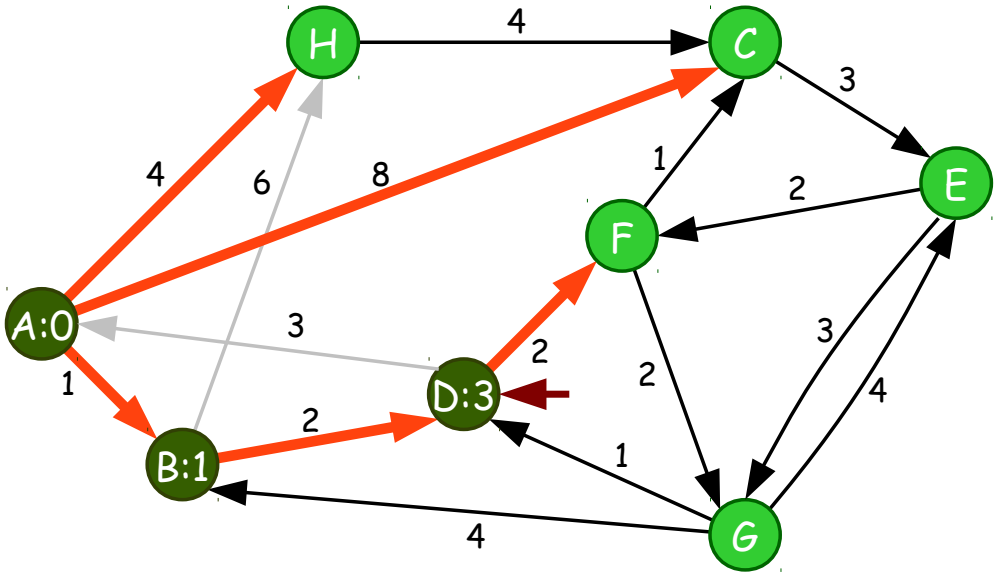


FILA

H	4
F	5
C	8
E	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

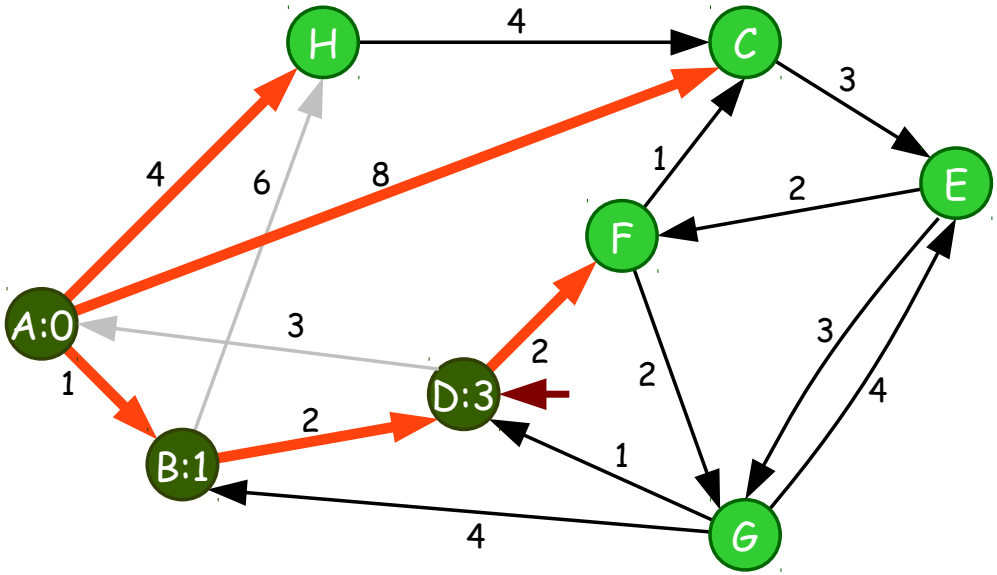


FILA

H	4
F	5
C	8
E	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

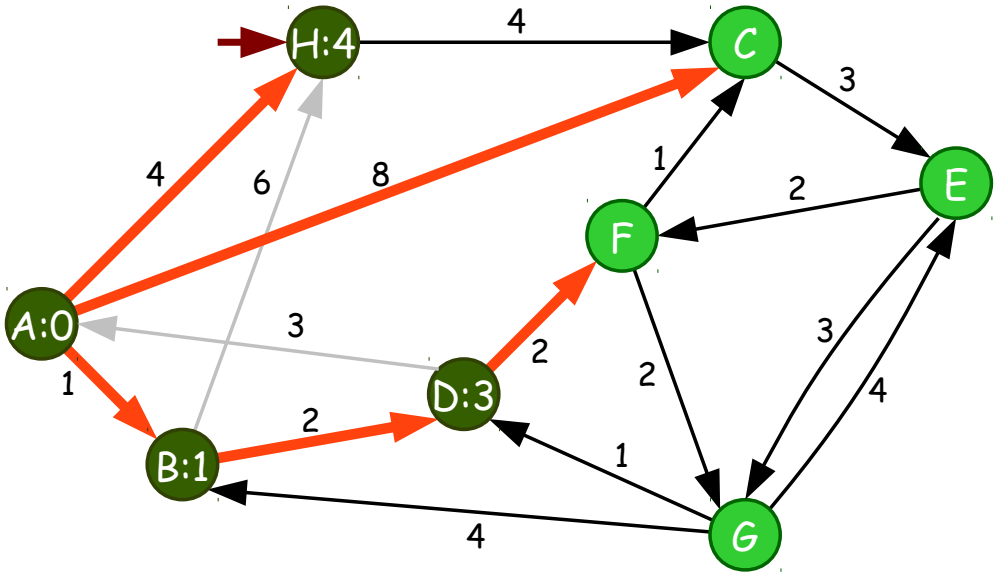


FILA

H	4
F	5
C	8
E	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

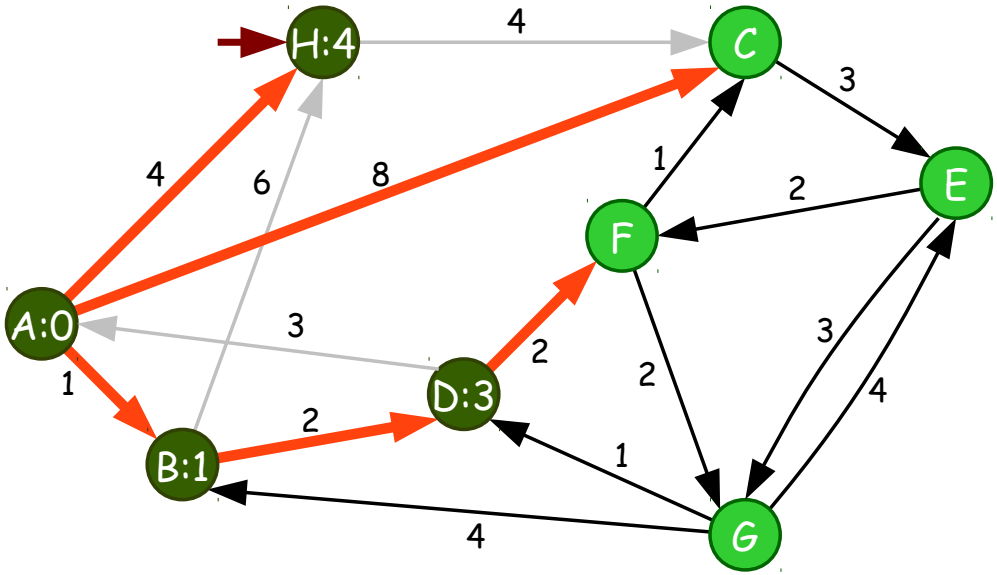


FILA

F	5
C	8
E	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

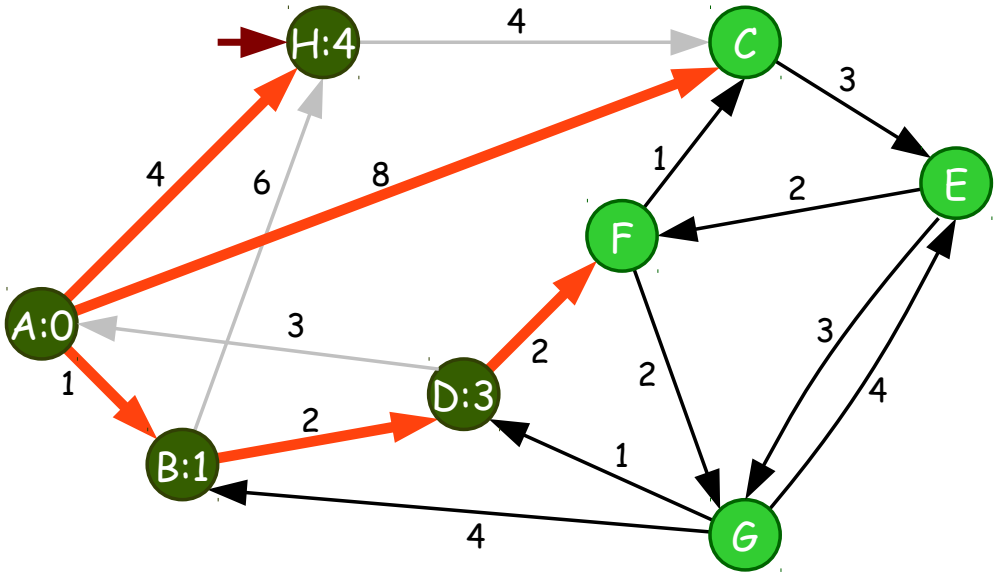


FILA

F	5
C	8
E	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

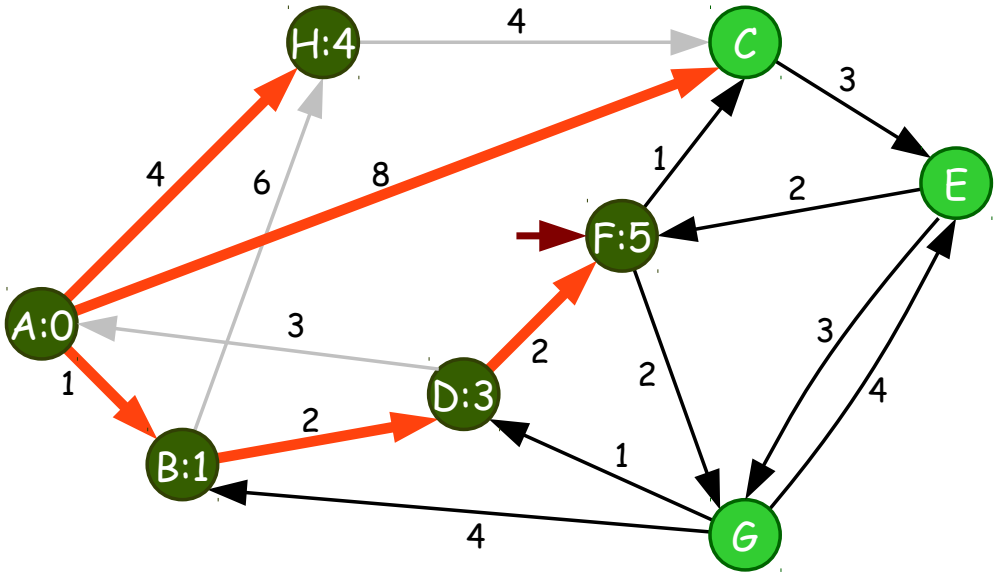


FILA

F	5
C	8
E	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

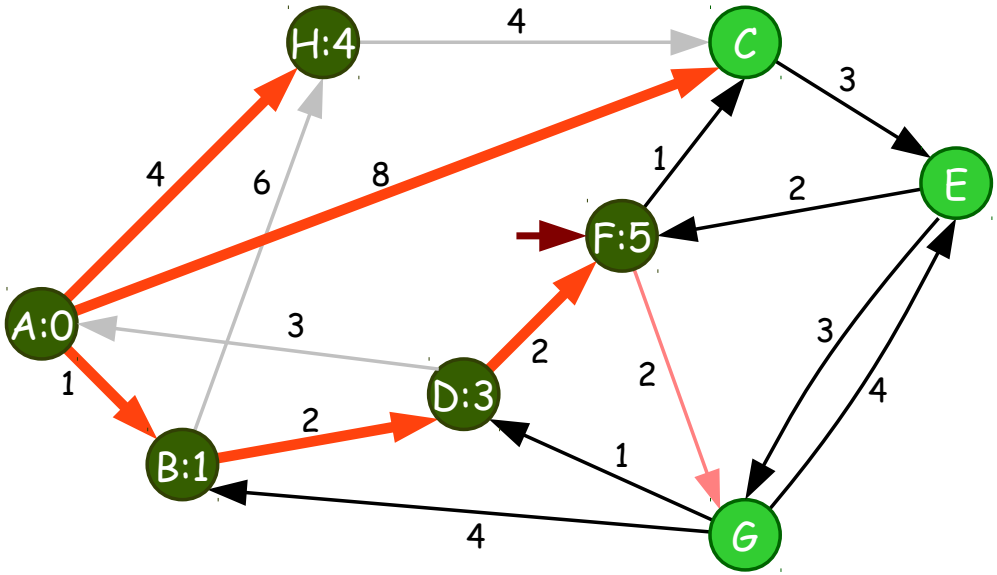


FILA

C	8
E	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

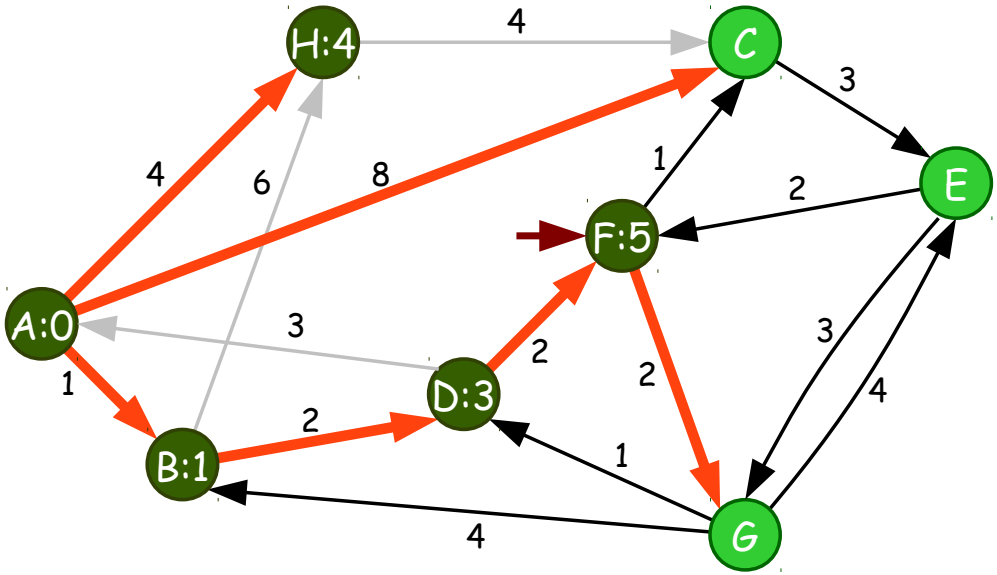


FILA

C	8
E	∞
G	∞

pular

Algoritmo de Dijkstra: exemplo

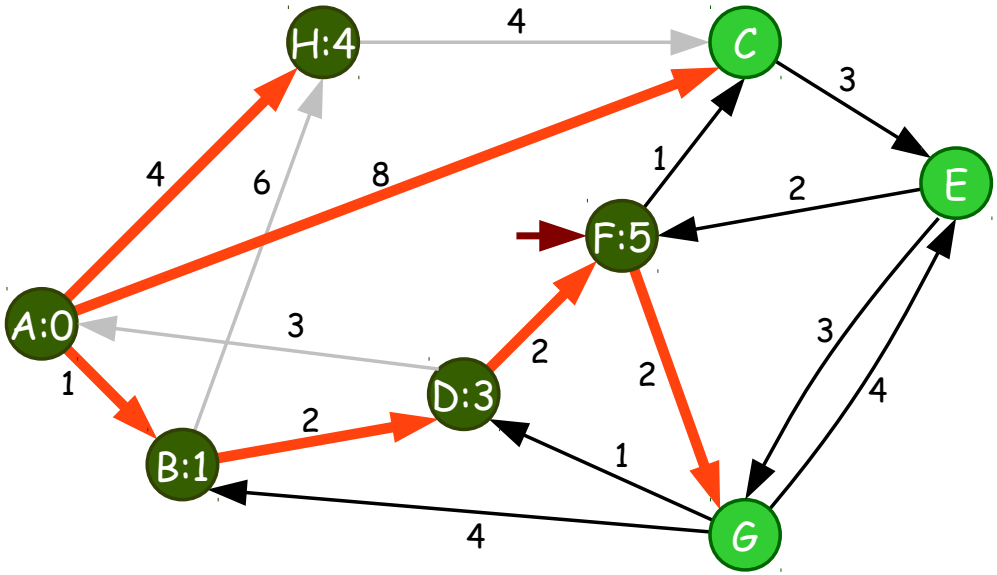


FILA

C	8
E	∞
G	7

pular

Algoritmo de Dijkstra: exemplo

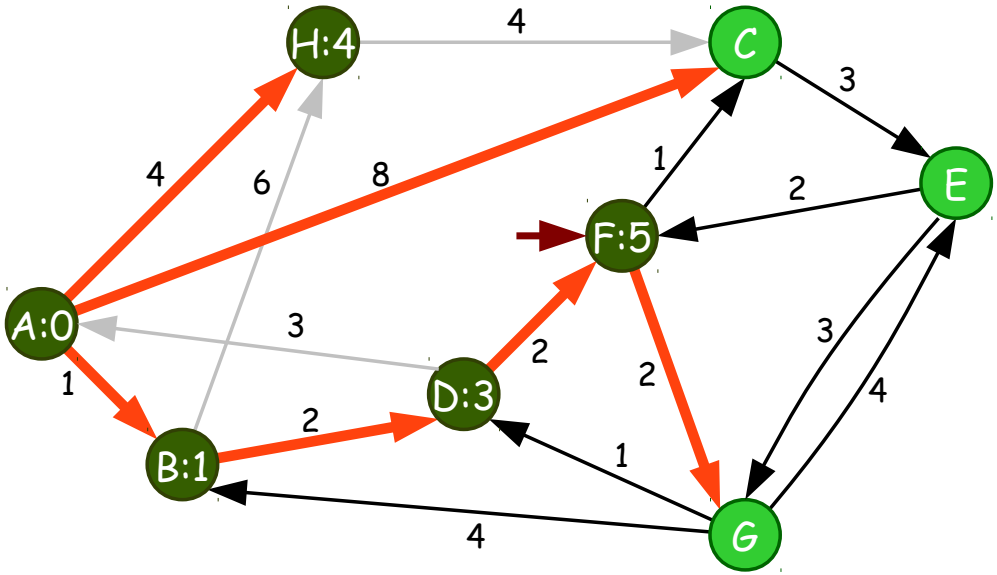


FILA

G	7
C	8
E	∞

pular

Algoritmo de Dijkstra: exemplo

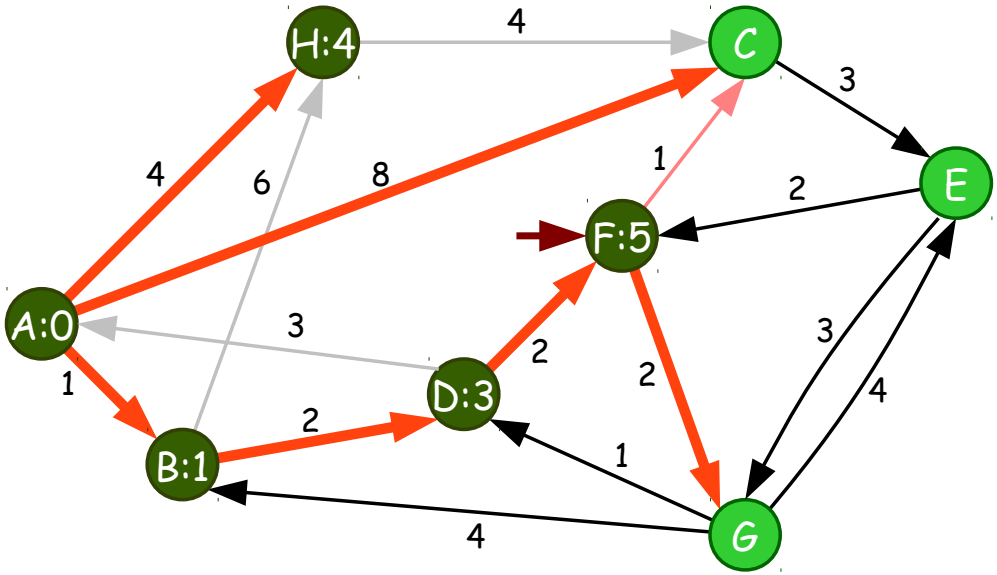


FILA

G	7
C	8
E	∞

pular

Algoritmo de Dijkstra: exemplo

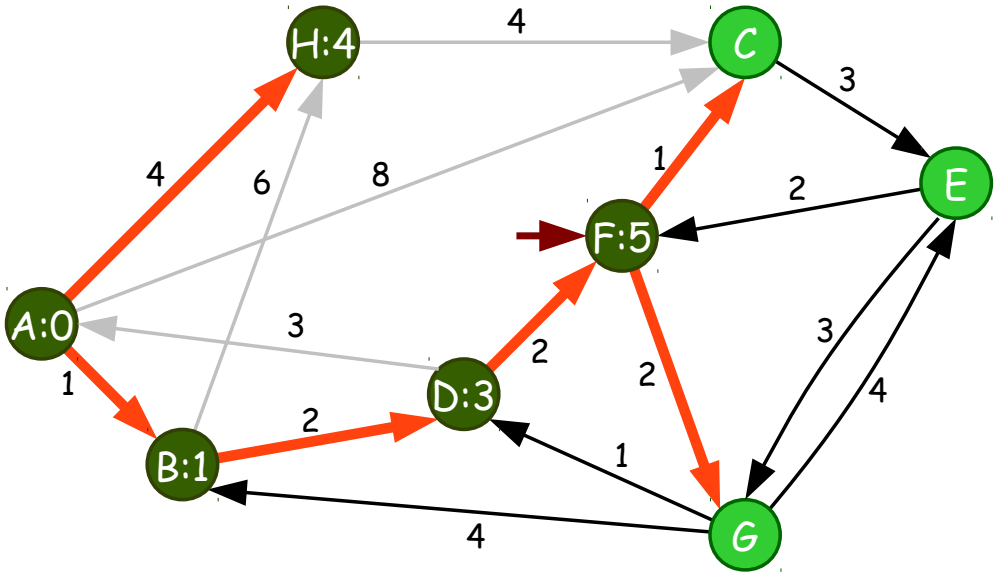


FILA

G	7
C	8
E	∞

pular

Algoritmo de Dijkstra: exemplo

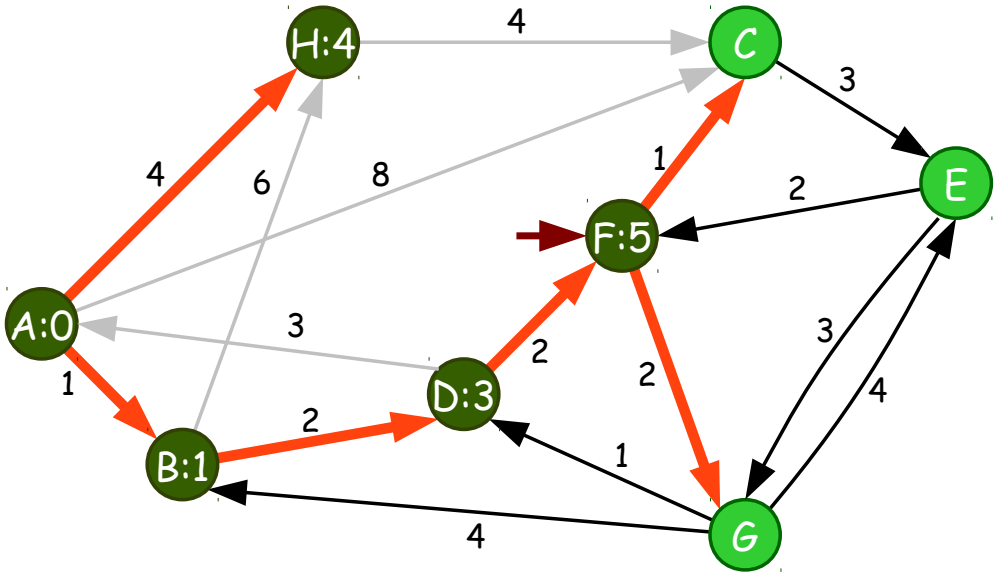


FILA

G	7
C	6
E	∞

pular

Algoritmo de Dijkstra: exemplo

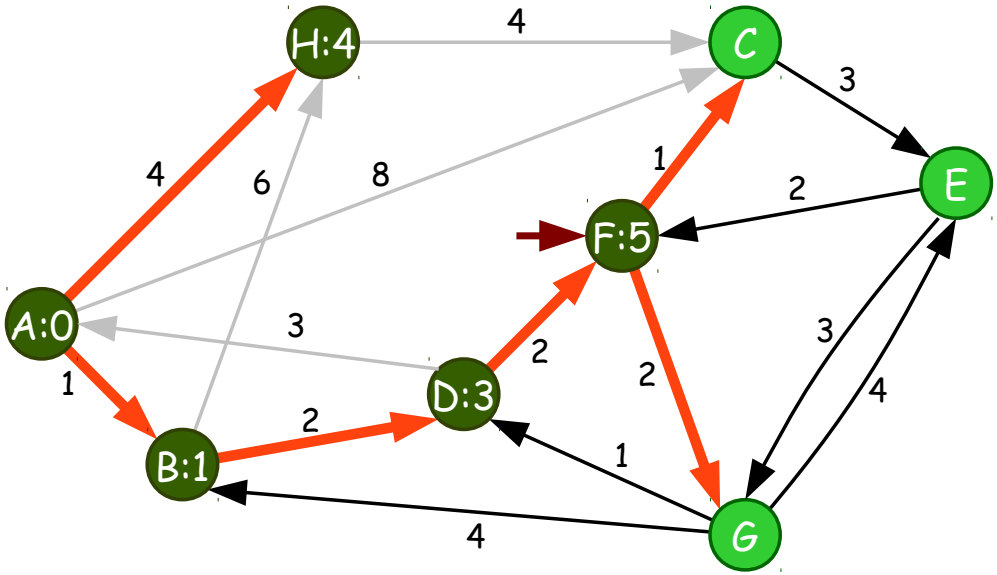


FILA

C	6
G	7
E	∞

pular

Algoritmo de Dijkstra: exemplo

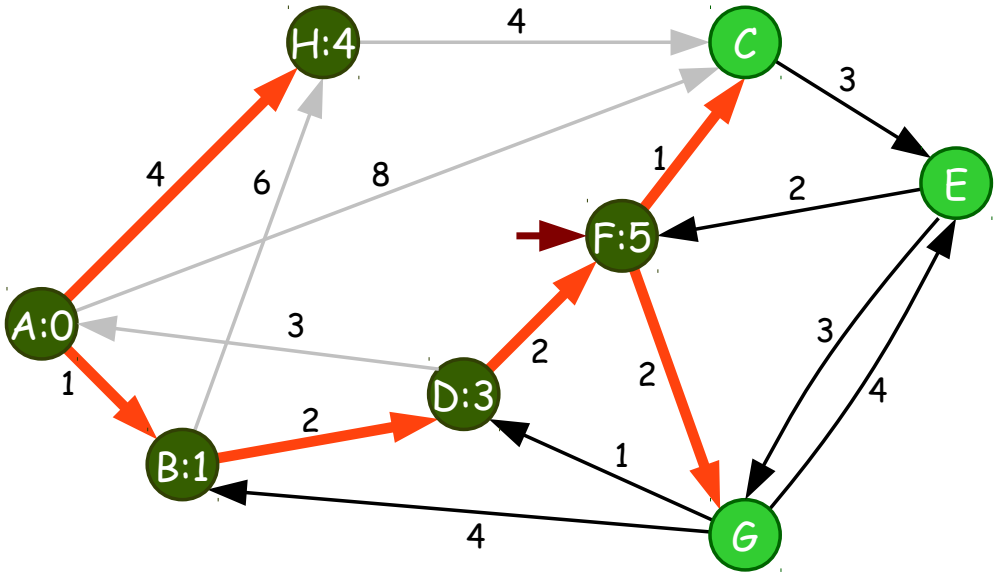


FILA

C	6
G	7
E	∞

pular

Algoritmo de Dijkstra: exemplo

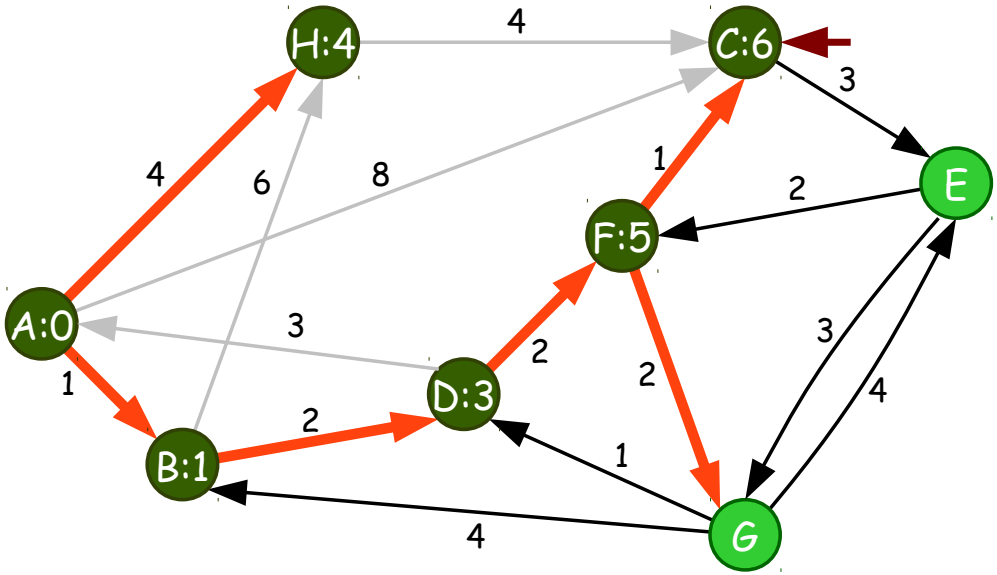


FILA

C	6
G	7
E	∞

pular

Algoritmo de Dijkstra: exemplo

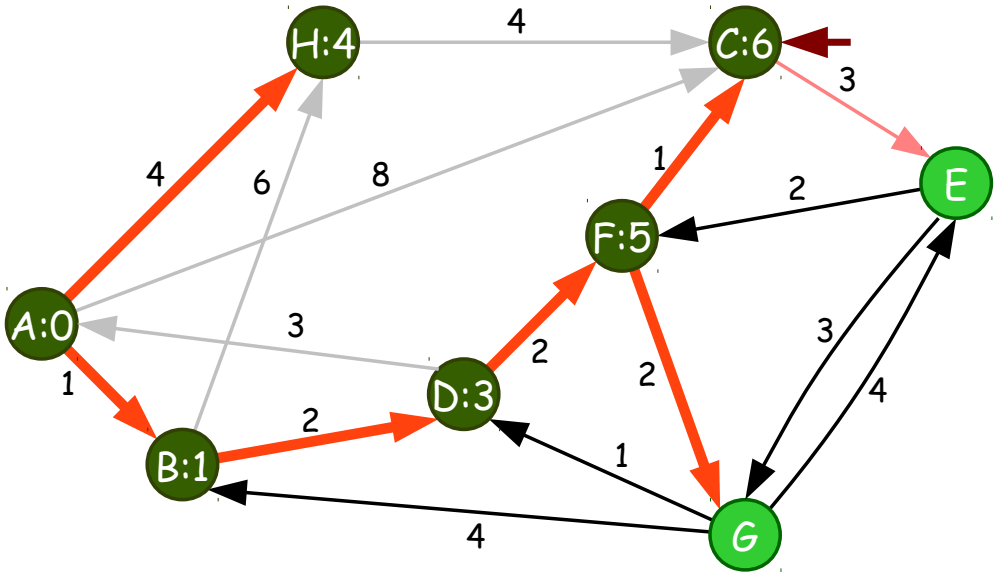


FILA

G	7
E	∞

pular

Algoritmo de Dijkstra: exemplo

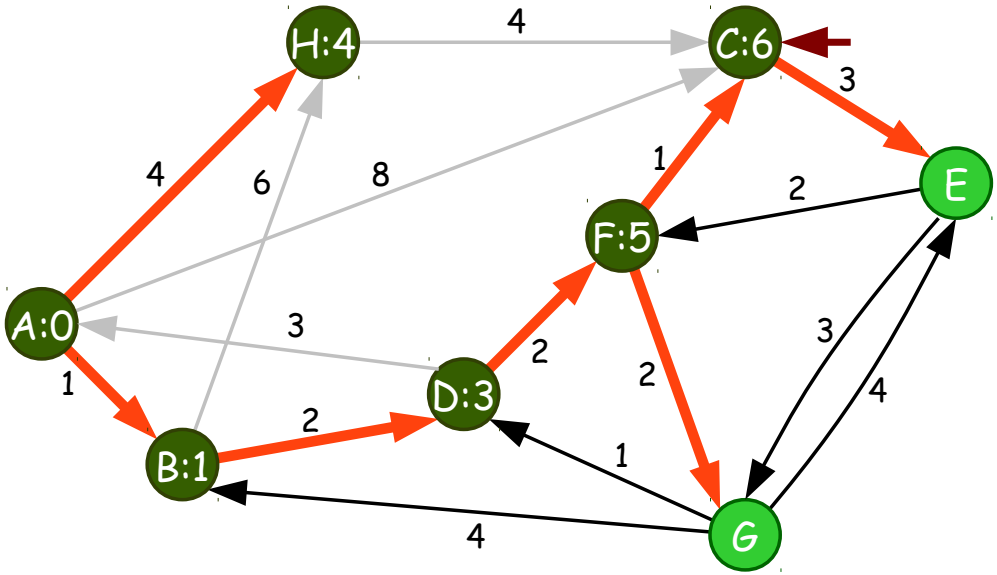


FILA

G	7
E	∞

pular

Algoritmo de Dijkstra: exemplo

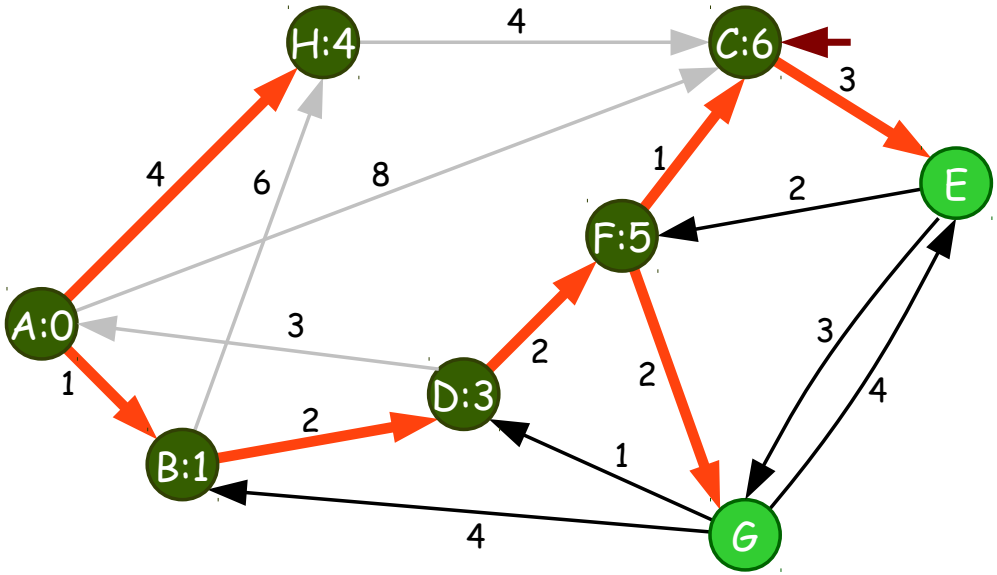


FILA

G	7
E	9

pular

Algoritmo de Dijkstra: exemplo

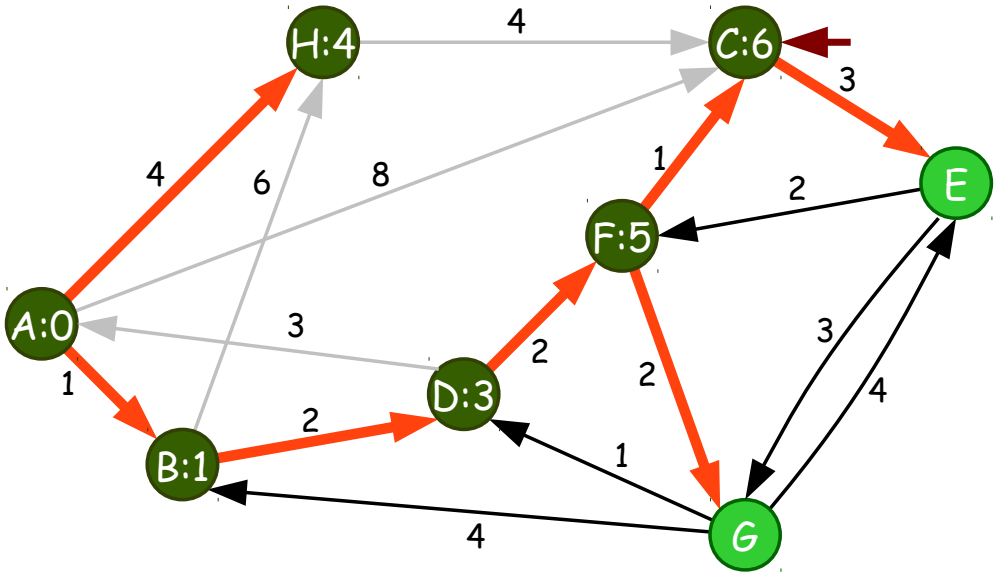


FILA

G	7
E	9

pular

Algoritmo de Dijkstra: exemplo

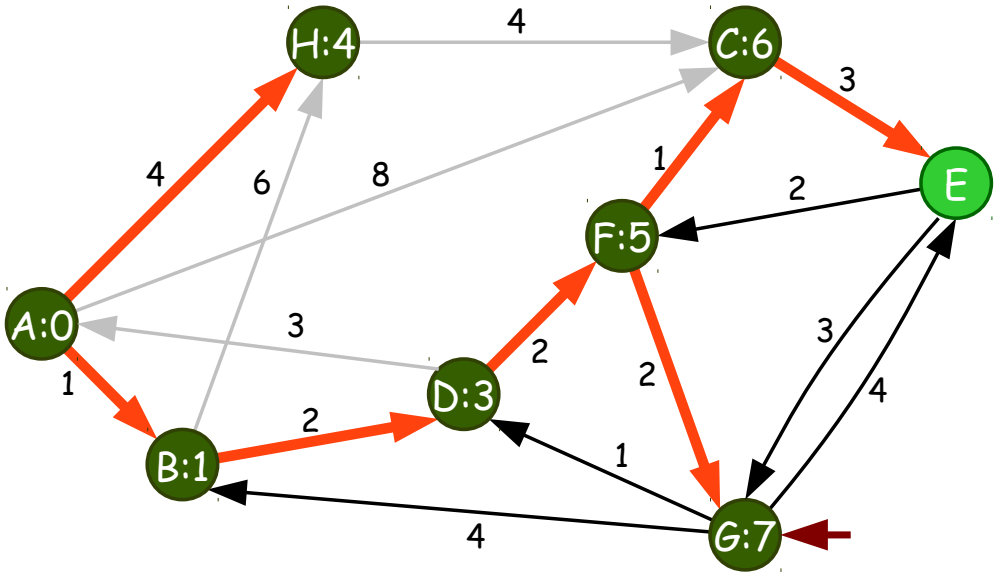


FILA

G	7
E	9

pular

Algoritmo de Dijkstra: exemplo

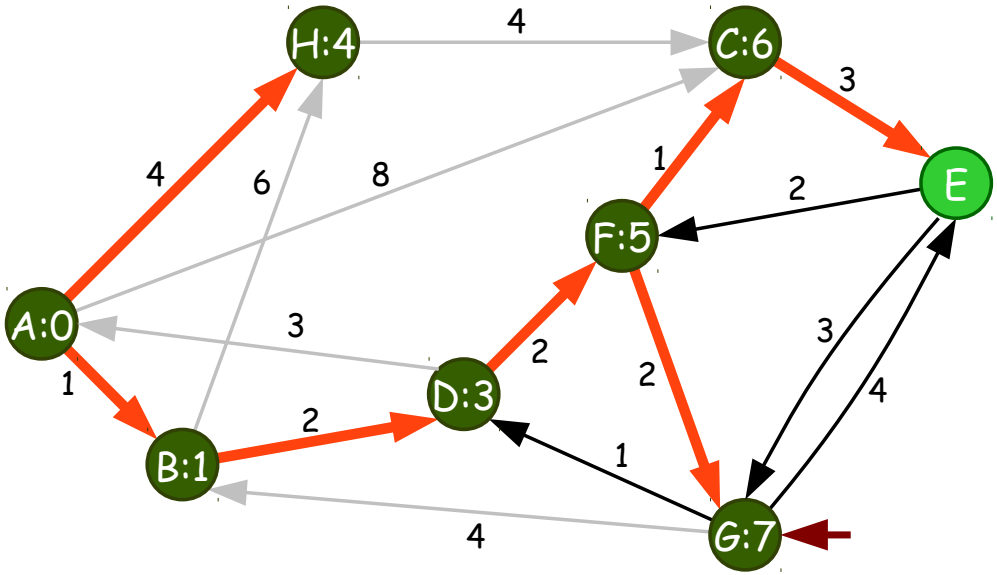


FILA

E 9

pular

Algoritmo de Dijkstra: exemplo

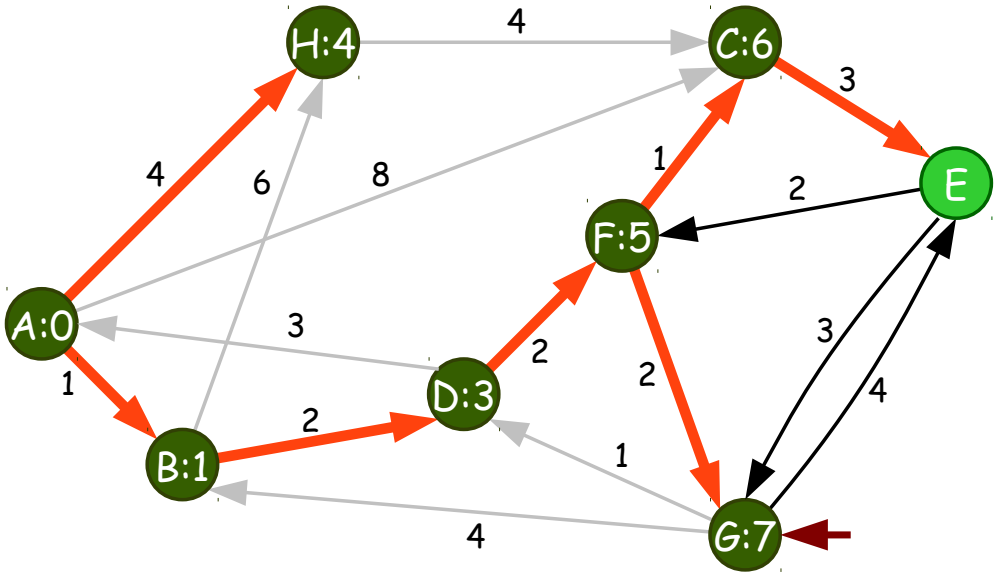


FILA

E 9

pular

Algoritmo de Dijkstra: exemplo

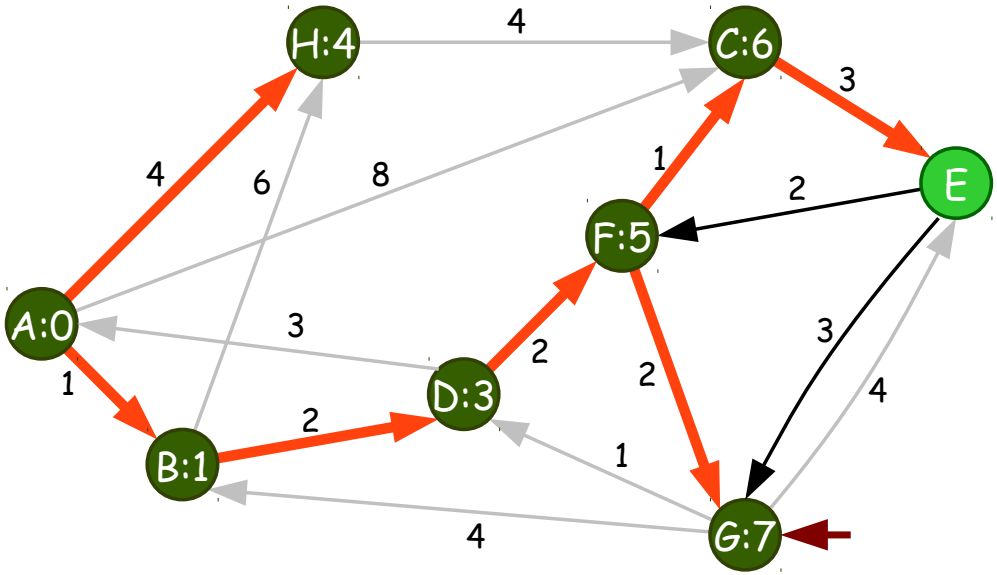


FILA

E 9

pular

Algoritmo de Dijkstra: exemplo

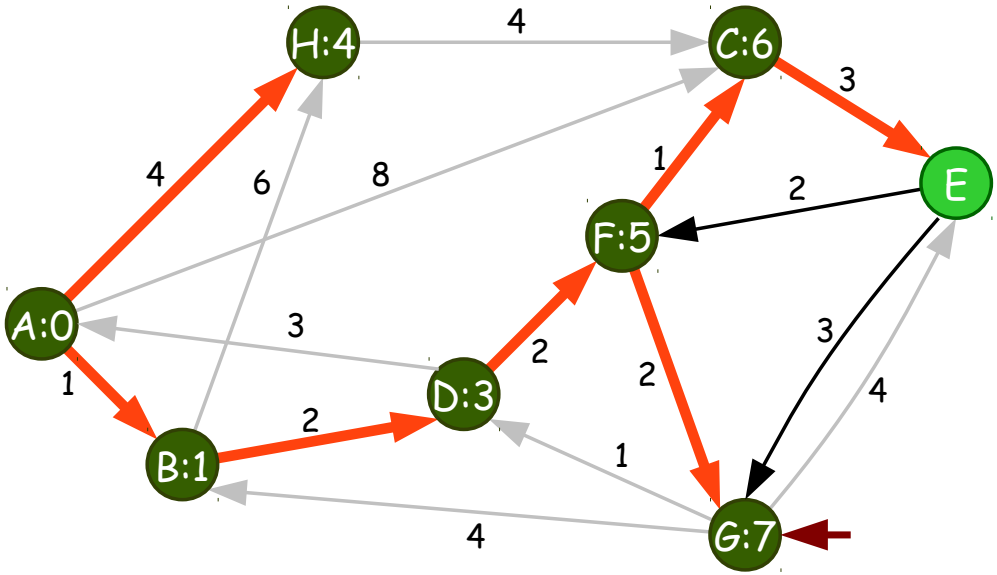


FILA

E 9

pular

Algoritmo de Dijkstra: exemplo

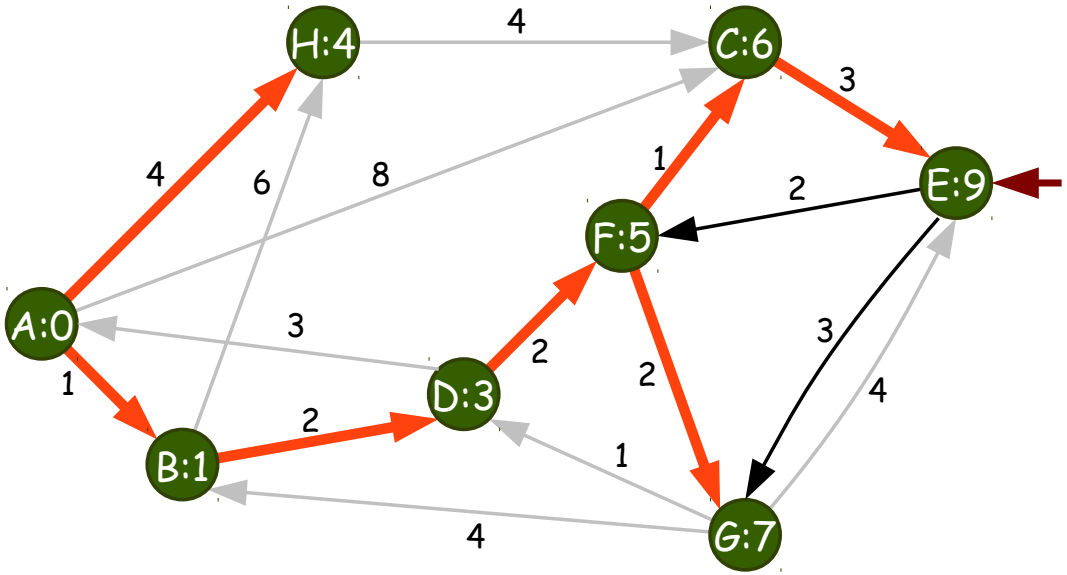


FILA

E 9

pular

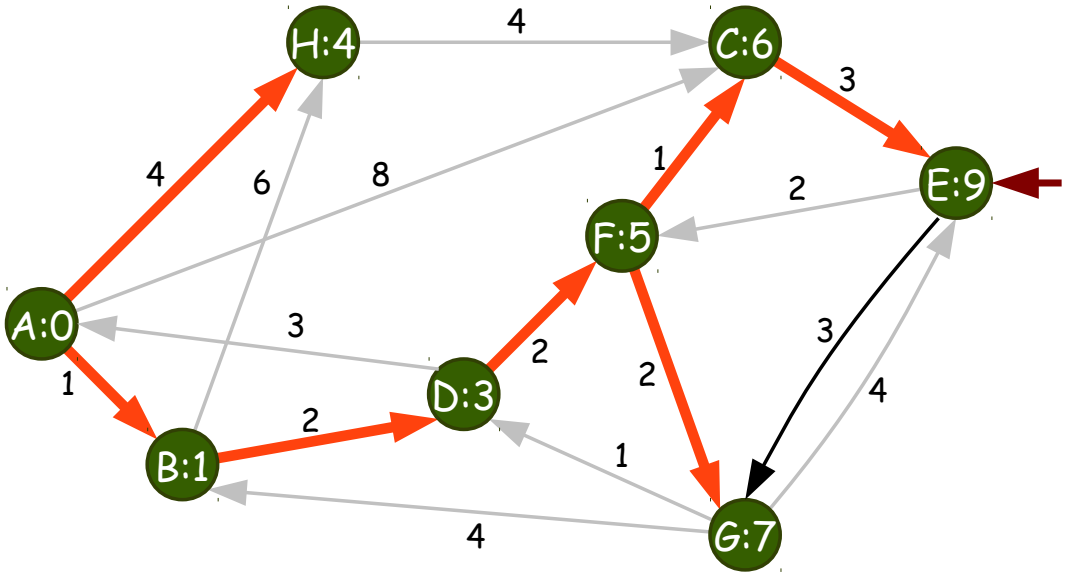
Algoritmo de Dijkstra: exemplo



FILA

pular

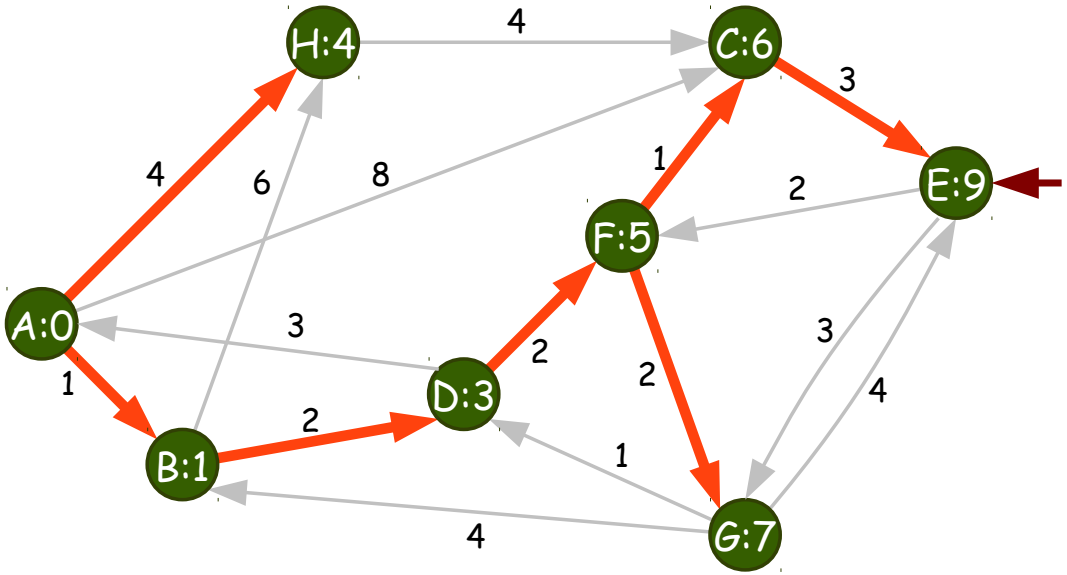
Algoritmo de Dijkstra: exemplo



FILA

pular

Algoritmo de Dijkstra: exemplo



FILA

pular

Exercício

1. O grafo do slide 12 contém ciclos. Modifique o algoritmos da ordenação topológica para verificar se um grafo tem ciclos.
2. Crie a árvore de caminhos mínimos para o grafo do slide 12 a partir do vértice D.
3. Implemente o algoritmo de Dijkstra em C.
4. A complexidade do algoritmo de Dijkstra depende da implementação da fila de prioridade usada. Utilizando um MinHeap, calcule a complexidade do algoritmo respondendo os seguintes itens:
   - Quanto tempo leva para criar um Heap?
   - Quantas vezes removemos o menor elemento?
   - Quantas vezes analisamos cada aresta?