

ALGORITMOS EM GRAFOS

Bacharelado em Sistemas da Informação

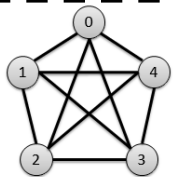
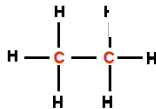
Prof. Marco André Abud Kappel

Aula 3 – Algoritmos em grafos

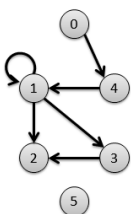
Algoritmos em grafos

- Anteriormente:

Grafos

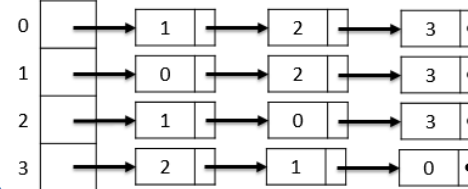


- Simples
- Completo
- Regular
- Cíclico
- Não direcionado
- Conexo



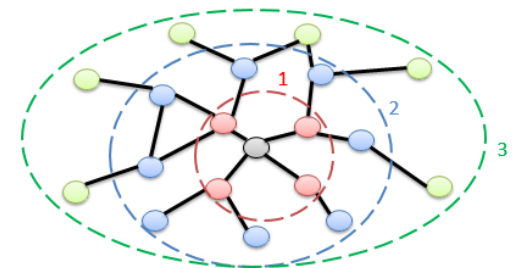
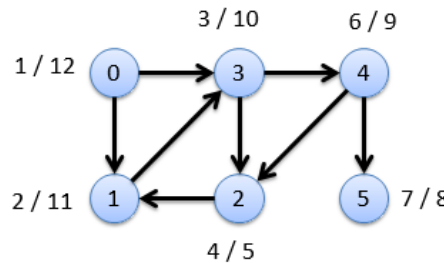
- Multigrafo
- Grafo direcionado
- Cíclico
- Grau médio 2,4
- Desconexo

Representações computacionais



	0	1	2	3	4	5
0	0	0	0	0	1	0
1	0	1	1	1	1	0
2	0	1	0	1	0	0
3	0	1	1	0	0	0
4	1	1	0	0	0	0
5	0	0	0	0	0	0

Percursos em grafos

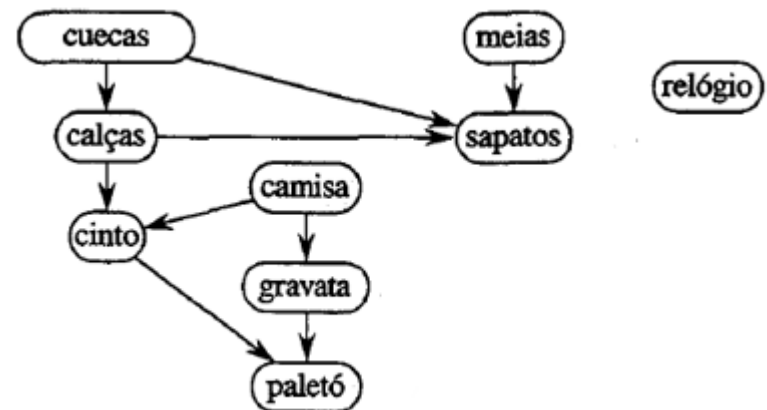
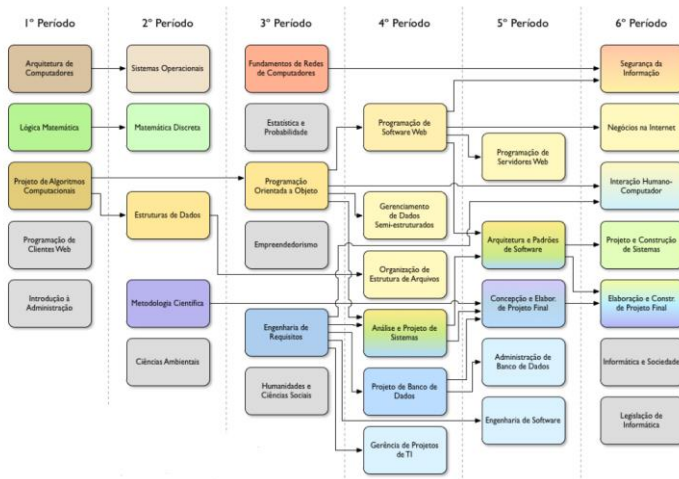


Hoje: Algoritmos em grafos

Algoritmos em grafos

- **Ordenação topológica**

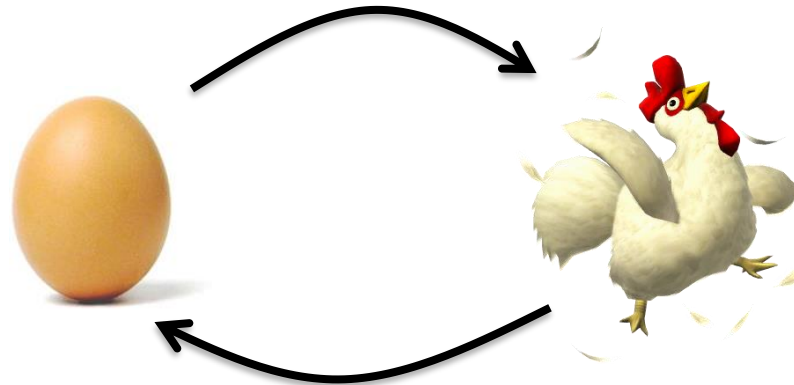
- Um conjunto contém **N tarefas** que precisam ser **executadas**.
- Essas tarefas são **dependentes**.
- Por exemplo, a **tarefa B** só pode ocorrer **depois** de **A**.
- **Pergunta:** Qual ordem de execução **não viola** as dependências?



Algoritmos em grafos

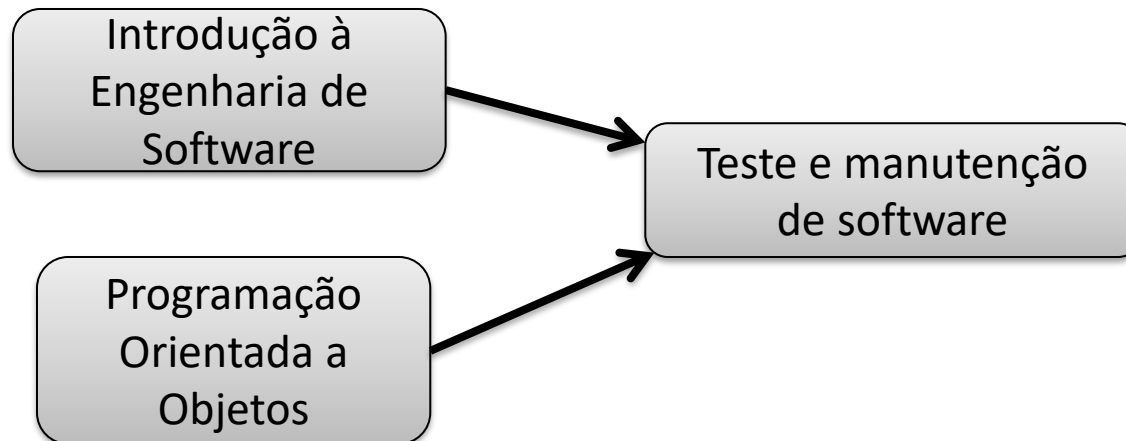
- **Ordenação topológica**

- O problema pode ser modelado por um **grafo direcionado** em que **vértices** são **tarefas** e **arestas direcionadas** indicam a **ordem de execução**.
- **Observação 1:** Impossível se o grafo for **cíclico**.



Algoritmos em grafos

- **Ordenação topológica**
 - **Observação 2:** Pode haver **mais de uma** solução.



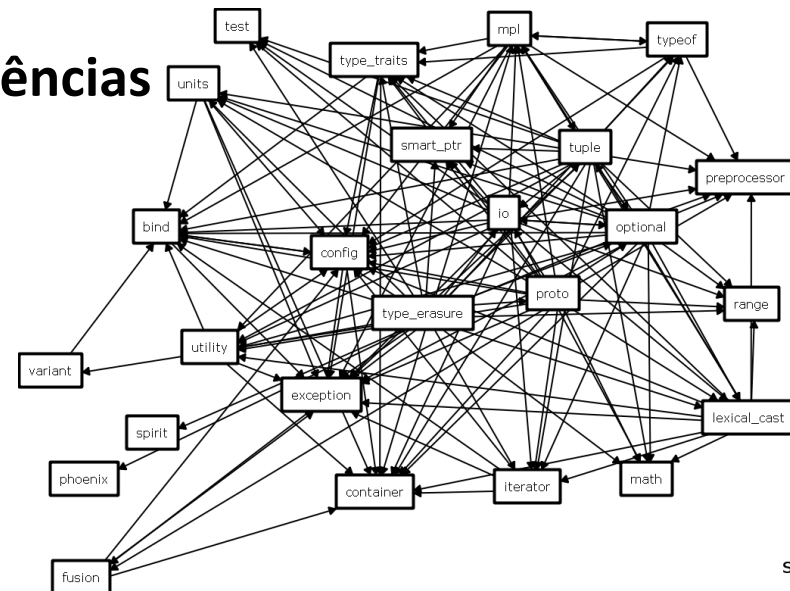
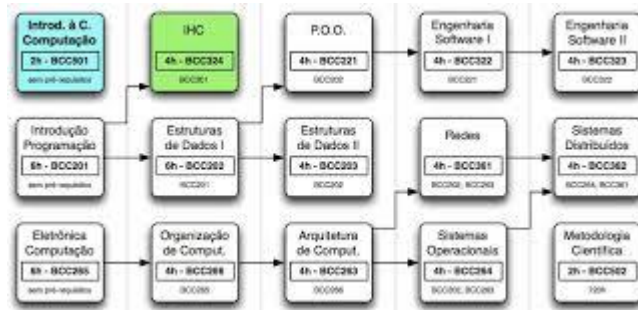
- É necessário cursar as duas disciplinas, mas a **ordem entre elas não importa**.

Algoritmos em grafos

- **Ordenação topológica**

- Exemplos de aplicações:

- 1) Dependência entre **tarefas triviais**
- 2) **Pré-requisitos** de disciplinas
- 3) Instalação de **pacotes com dependências**
- 4) **Compilação** de sistemas



Algoritmos em grafos

- **Ordenação topológica**

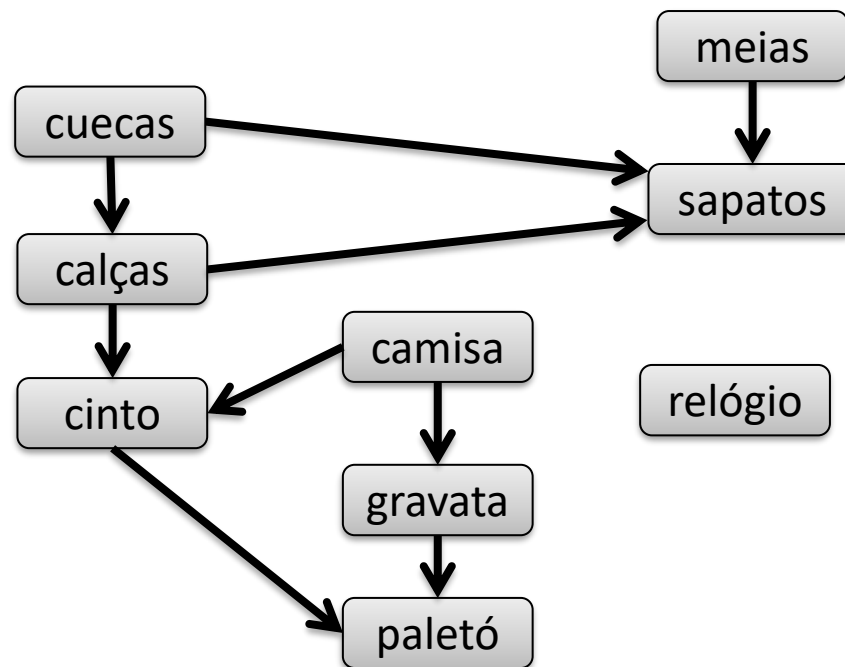
- **Algoritmo:** Utilizar a **busca em profundidade** para percorrer todos os nós!
 1. Primeiro, é escolhido um **nó sem dependências**.
 2. A partir deste nó, é aplicada a **busca em profundidade**.
 3. À medida que cada vértice é **terminado**, inserir no **início de uma lista encadeada**.
 4. Se ainda restam **nós não visitados**, retornar ao passo 1.
- A **lista encadeada** resultante representa a **ordenação topológica** do grafo, ou seja, uma **sequência de execução válida** para todas as tarefas.



Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

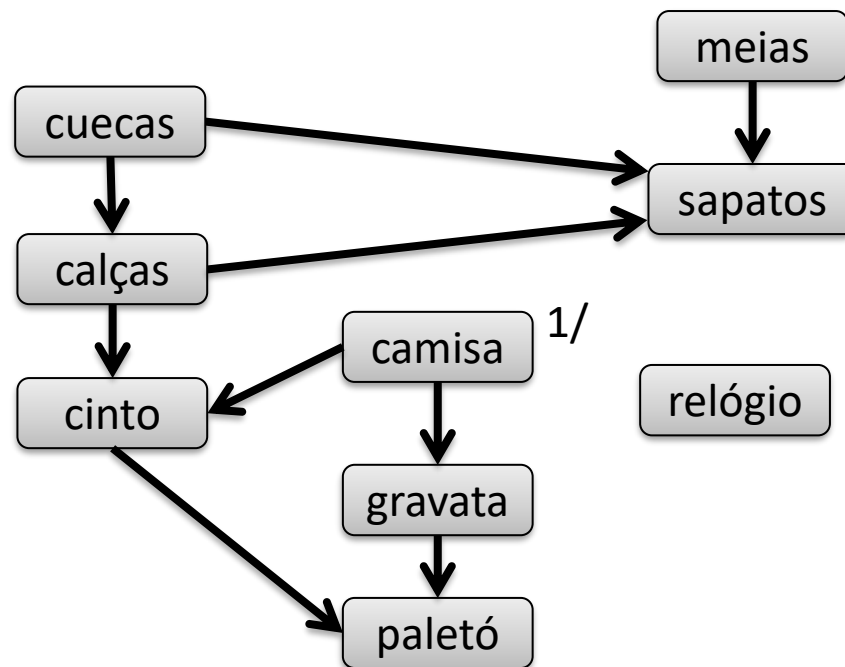


Ordenação topológica:

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

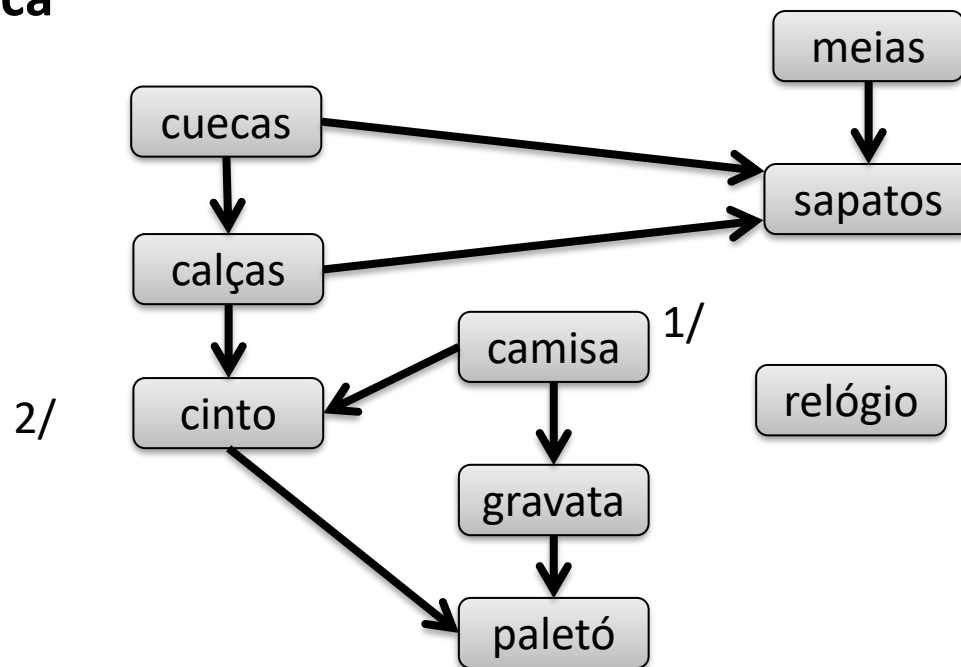


Ordenação topológica:

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

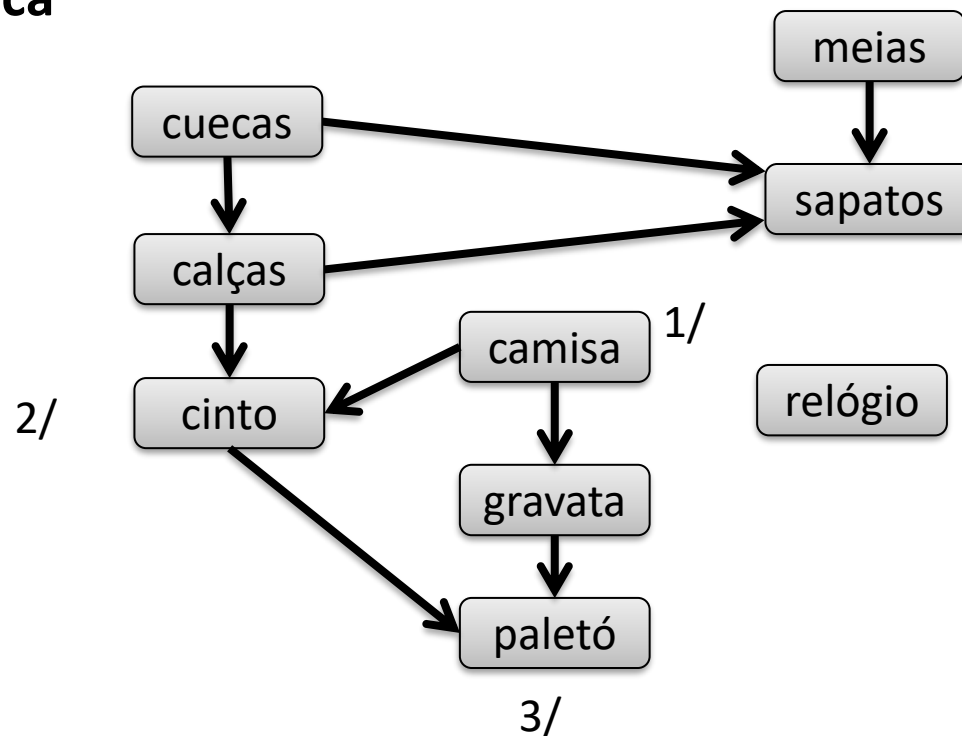


Ordenação topológica:

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

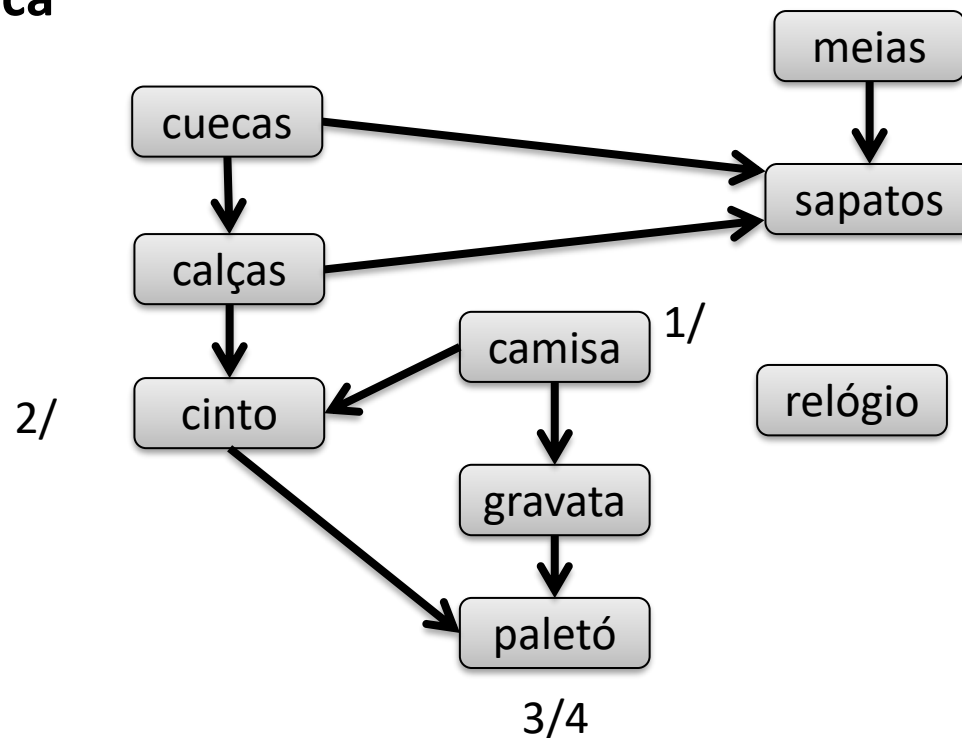


Ordenação topológica:

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



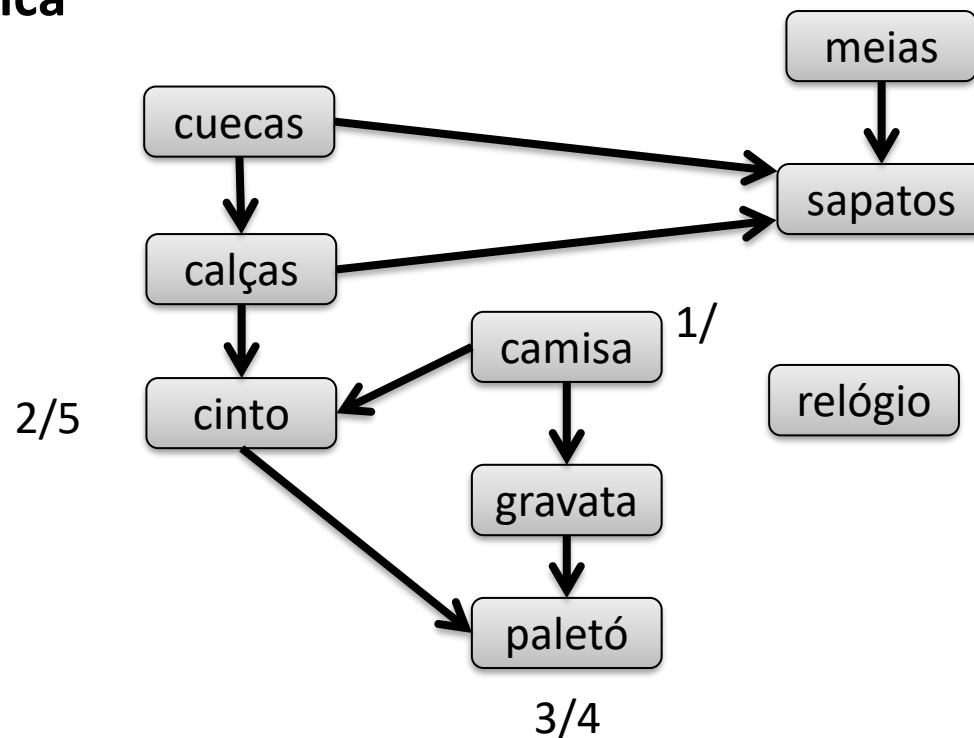
Ordenação topológica:

paletó

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



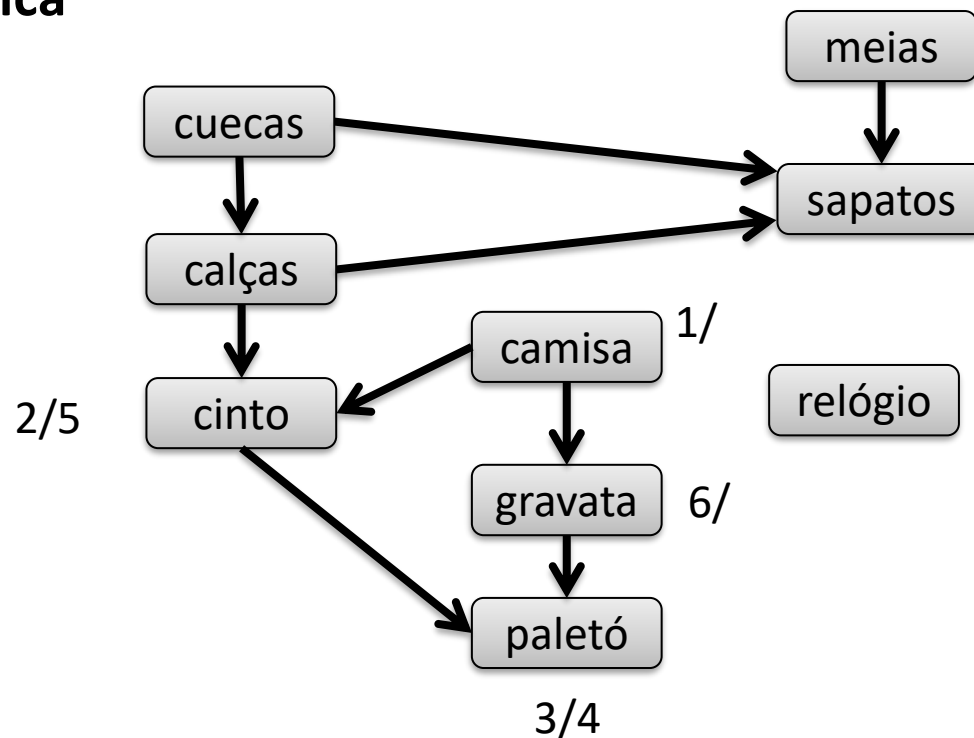
Ordenação topológica:



Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



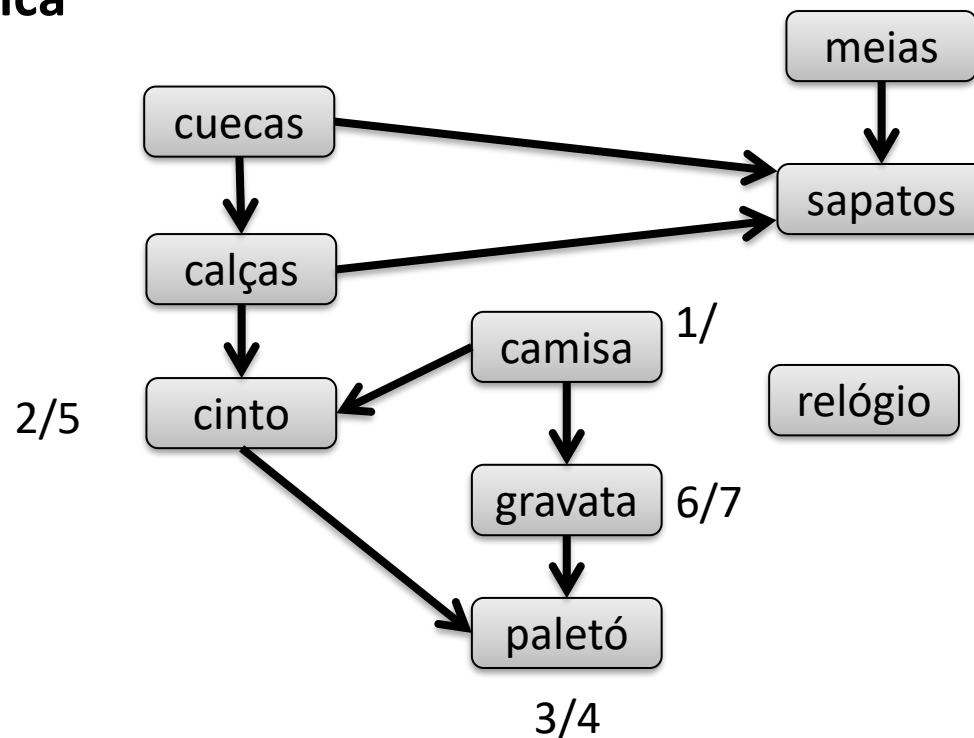
Ordenação topológica:



Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

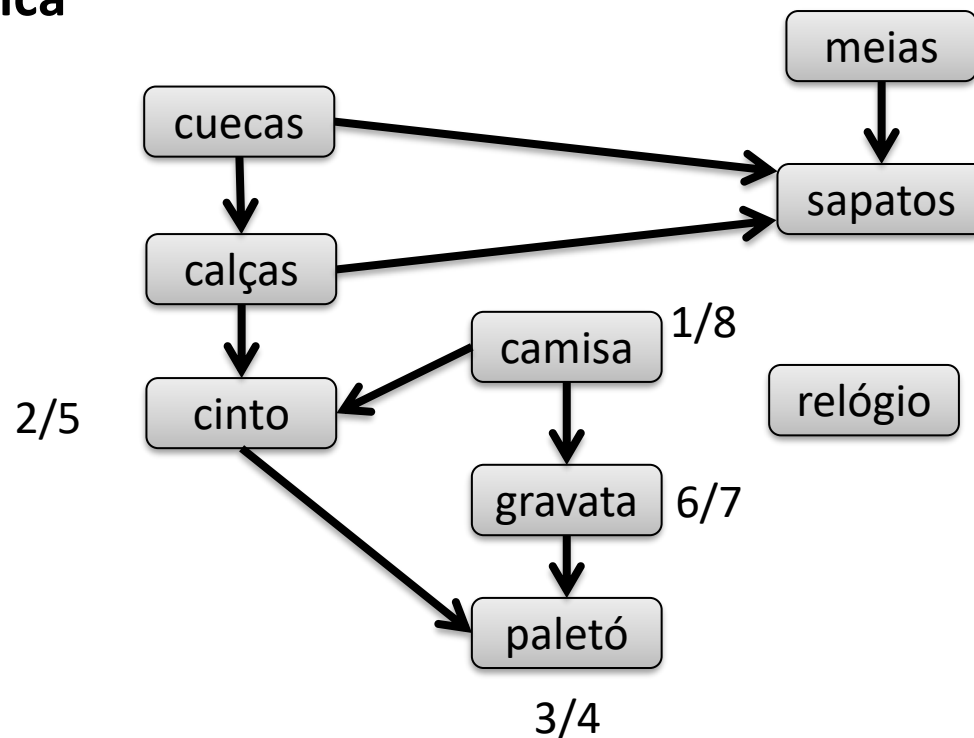


Ordenação topológica: gravata → cinto → paletó

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

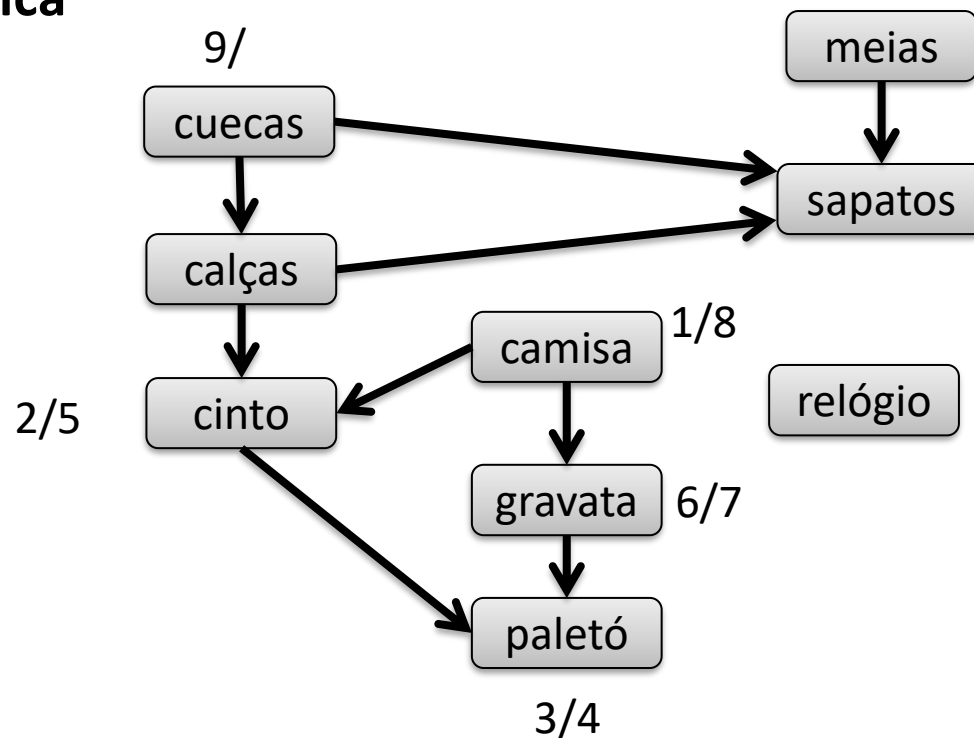


Ordenação topológica: camisa → gravata → cinto → paletó

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

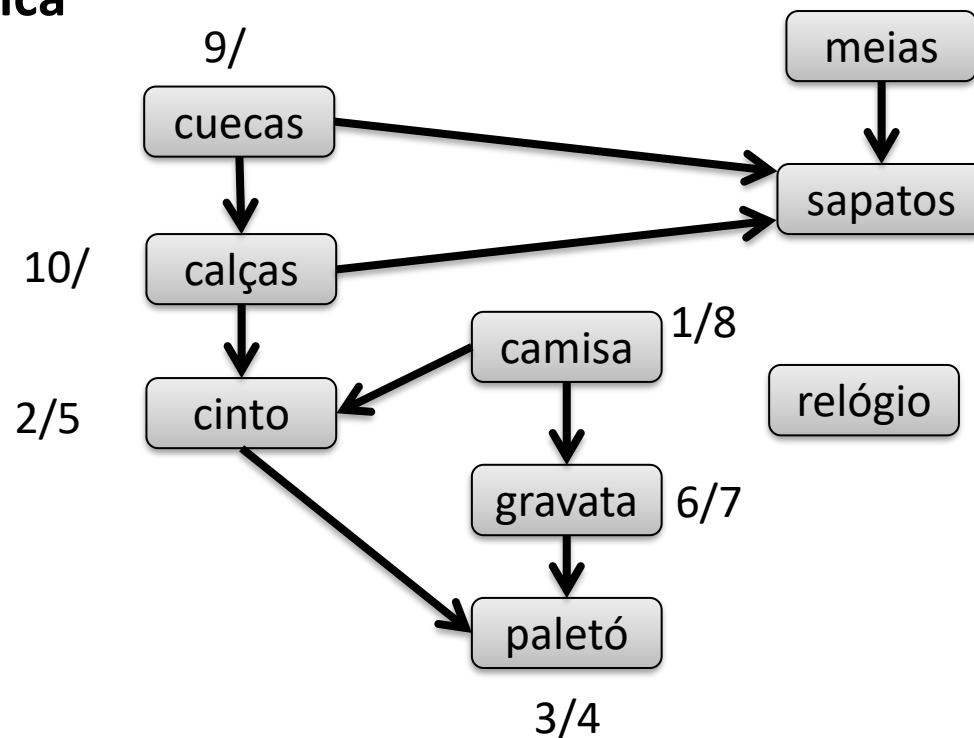


Ordenação topológica: camisa → gravata → cinto → paletó

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



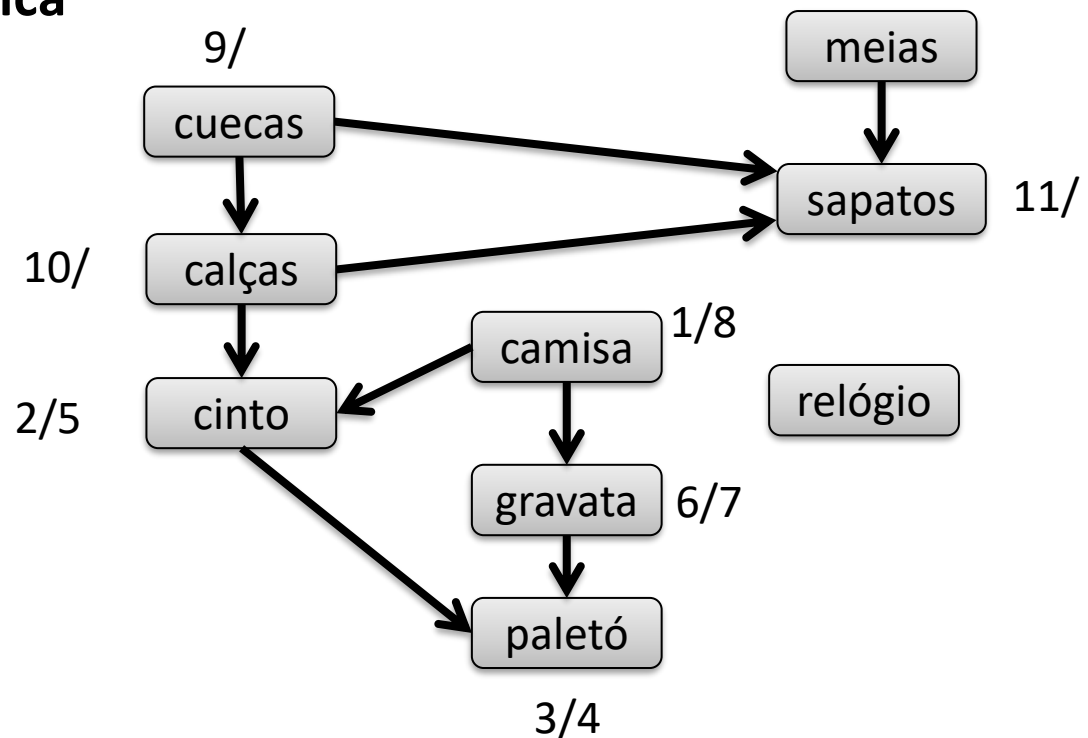
Ordenação topológica:



Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

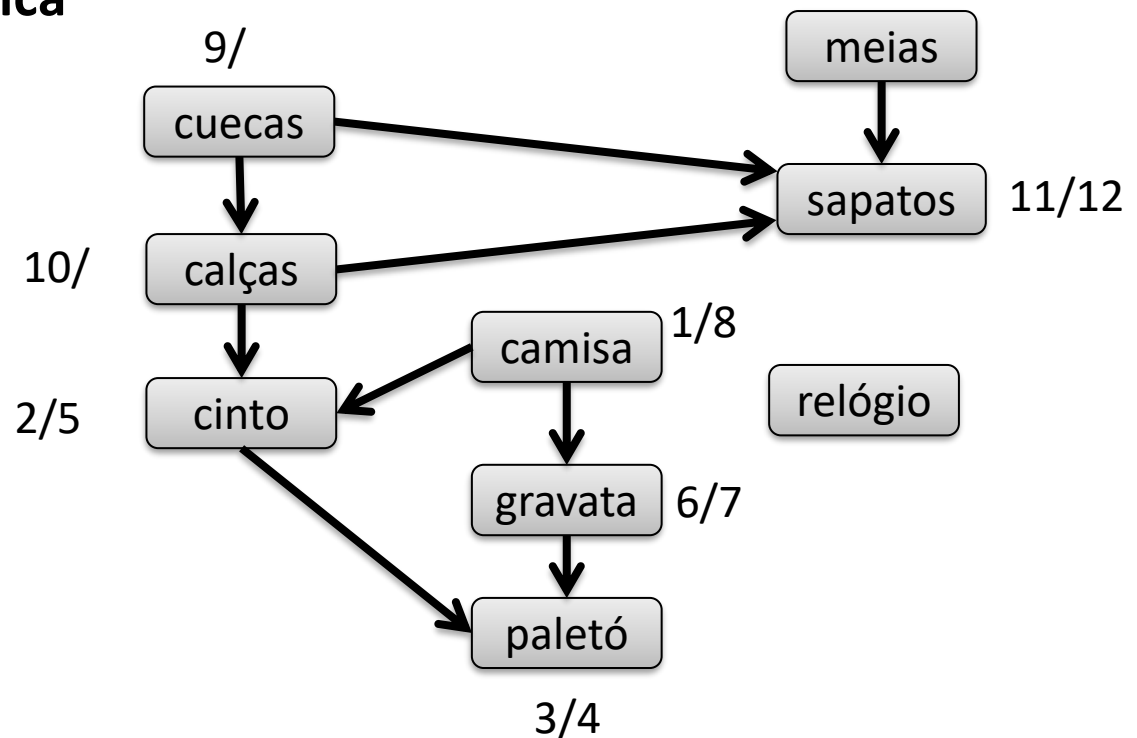


Ordenação topológica: camisa → gravata → cinto → paletó

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



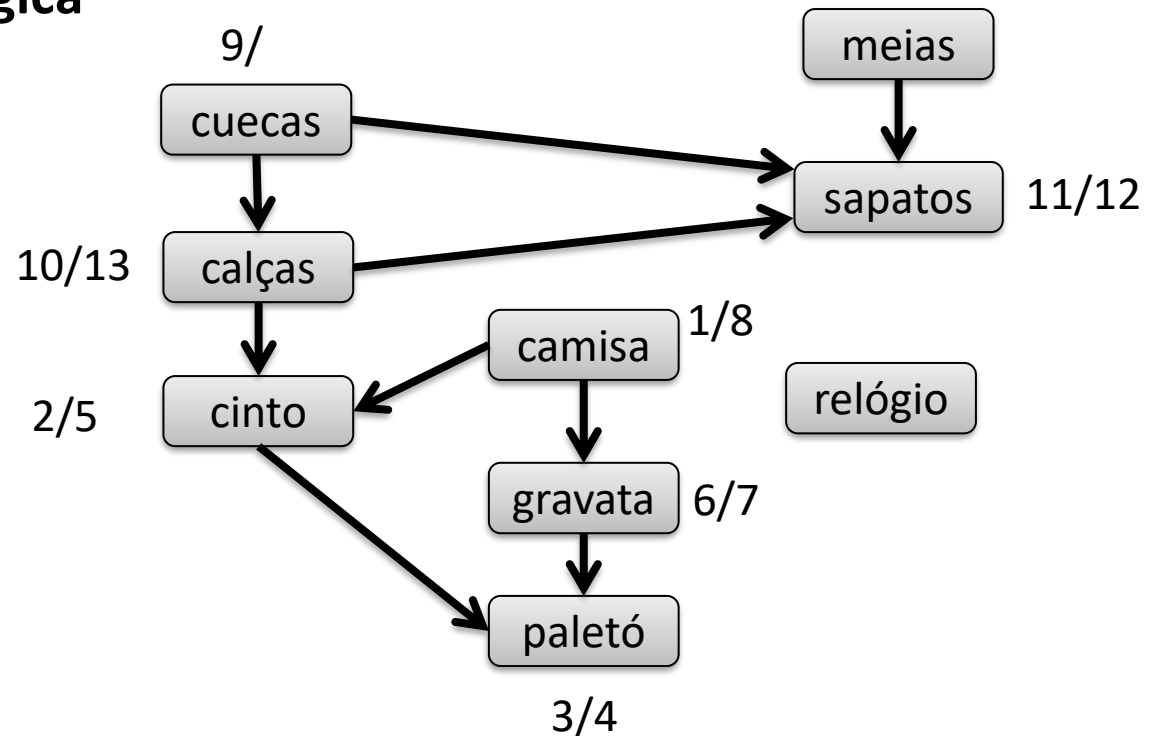
Ordenação topológica:



Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:

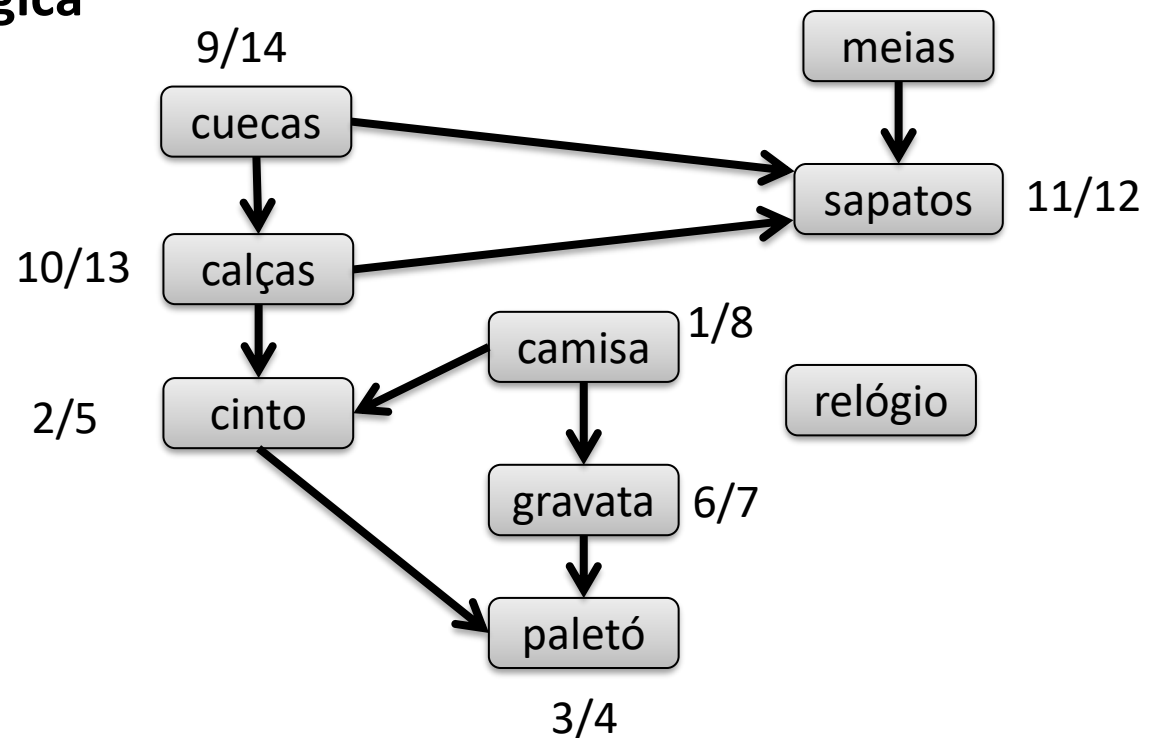


Ordenação topológica: calças → sapatos → camisa → gravata → cinto → paletó

Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



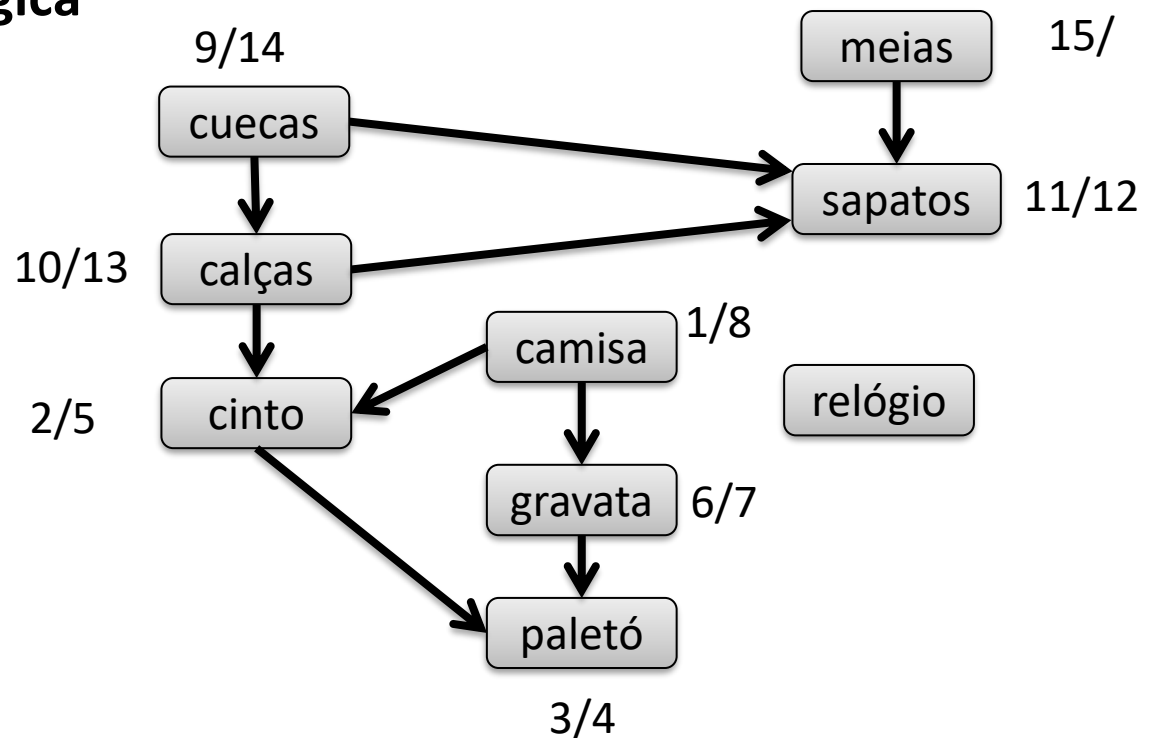
Ordenação topológica:



Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



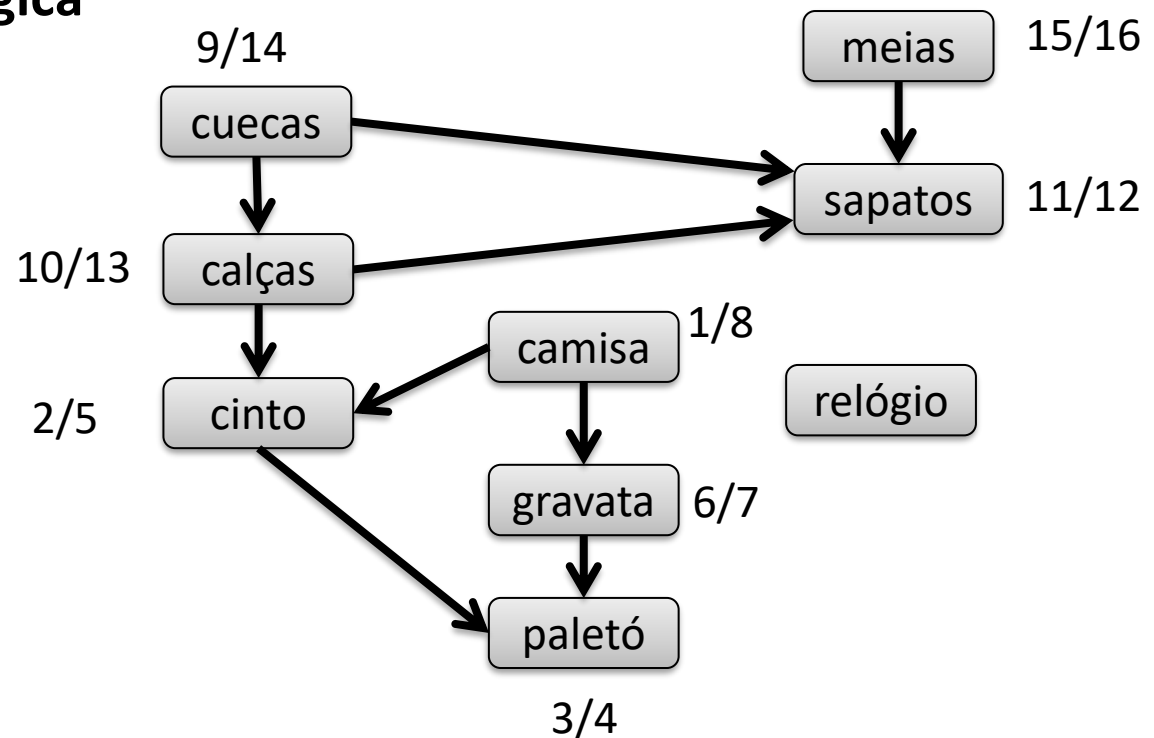
Ordenação topológica:



Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



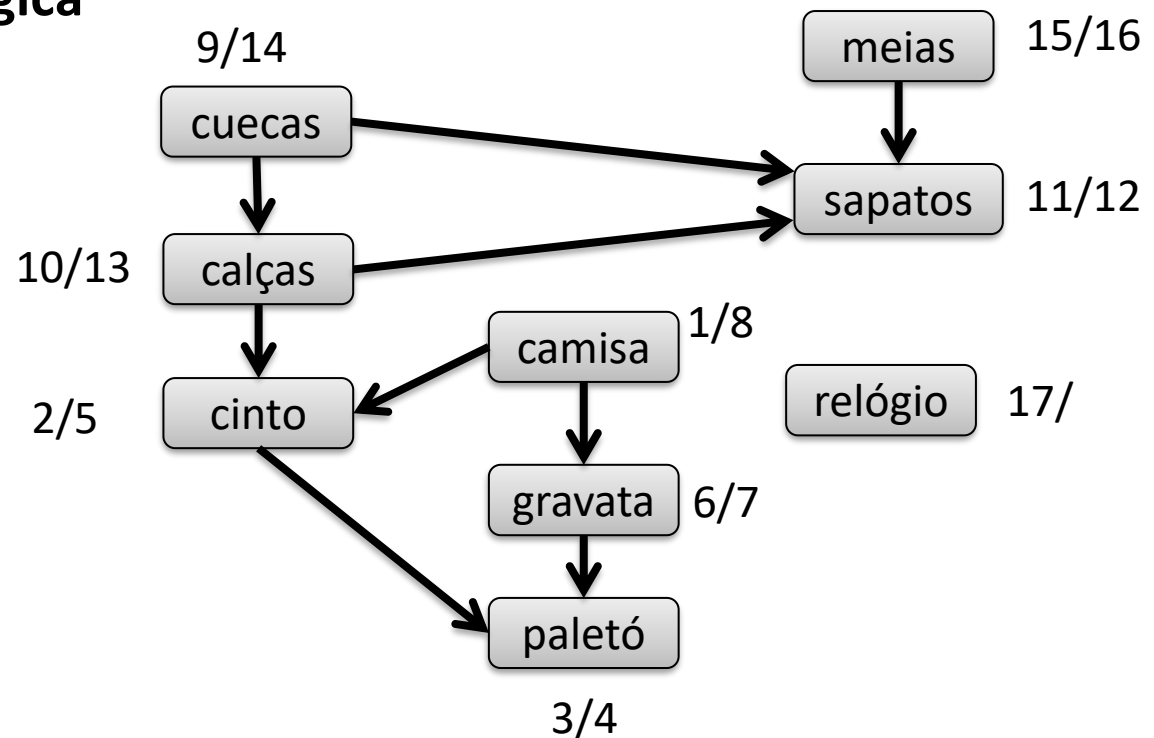
Ordenação topológica:



Algoritmos em grafos

- **Ordenação topológica**

- Exemplo:



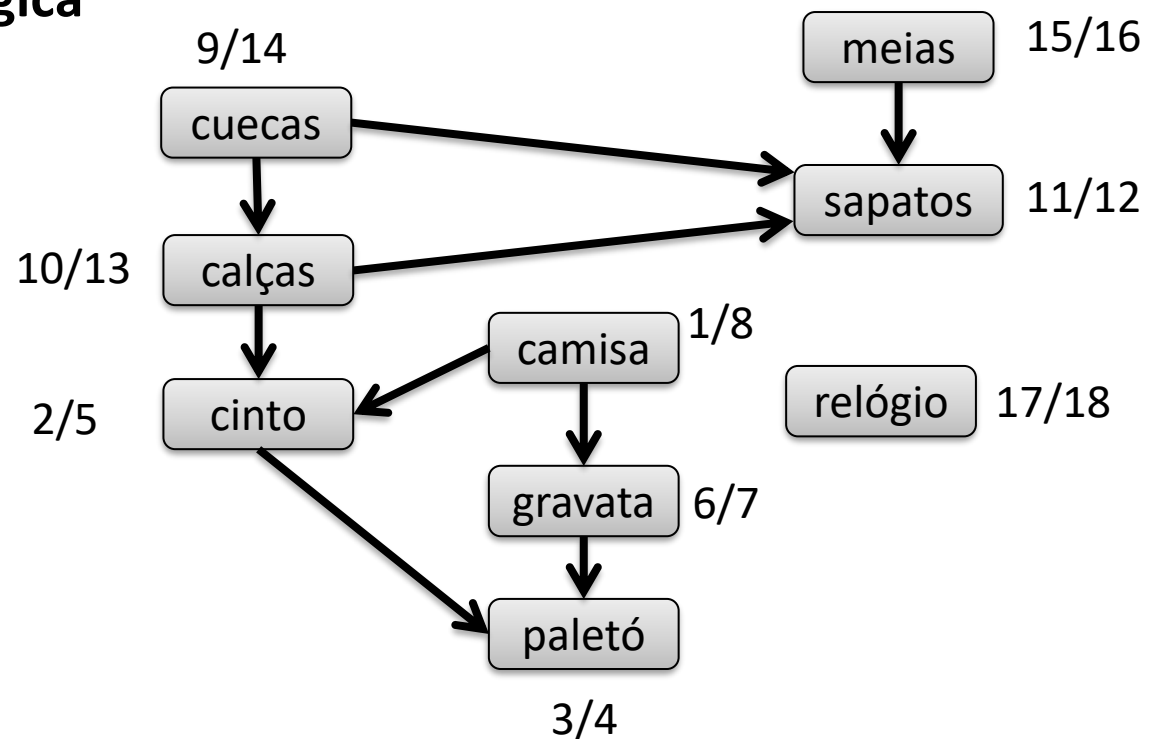
Ordenação topológica:



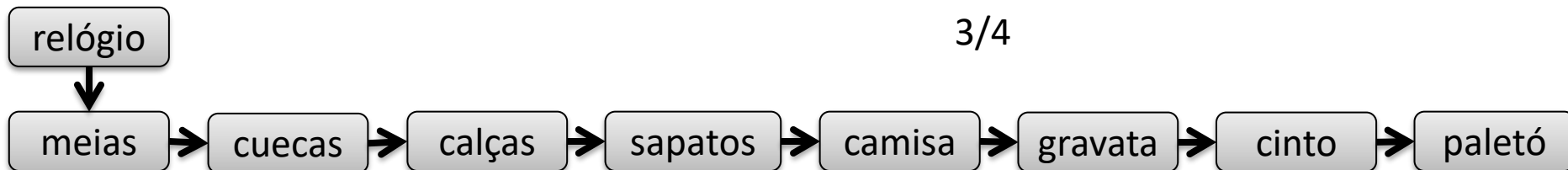
Algoritmos em grafos

- **Ordenação topológica**

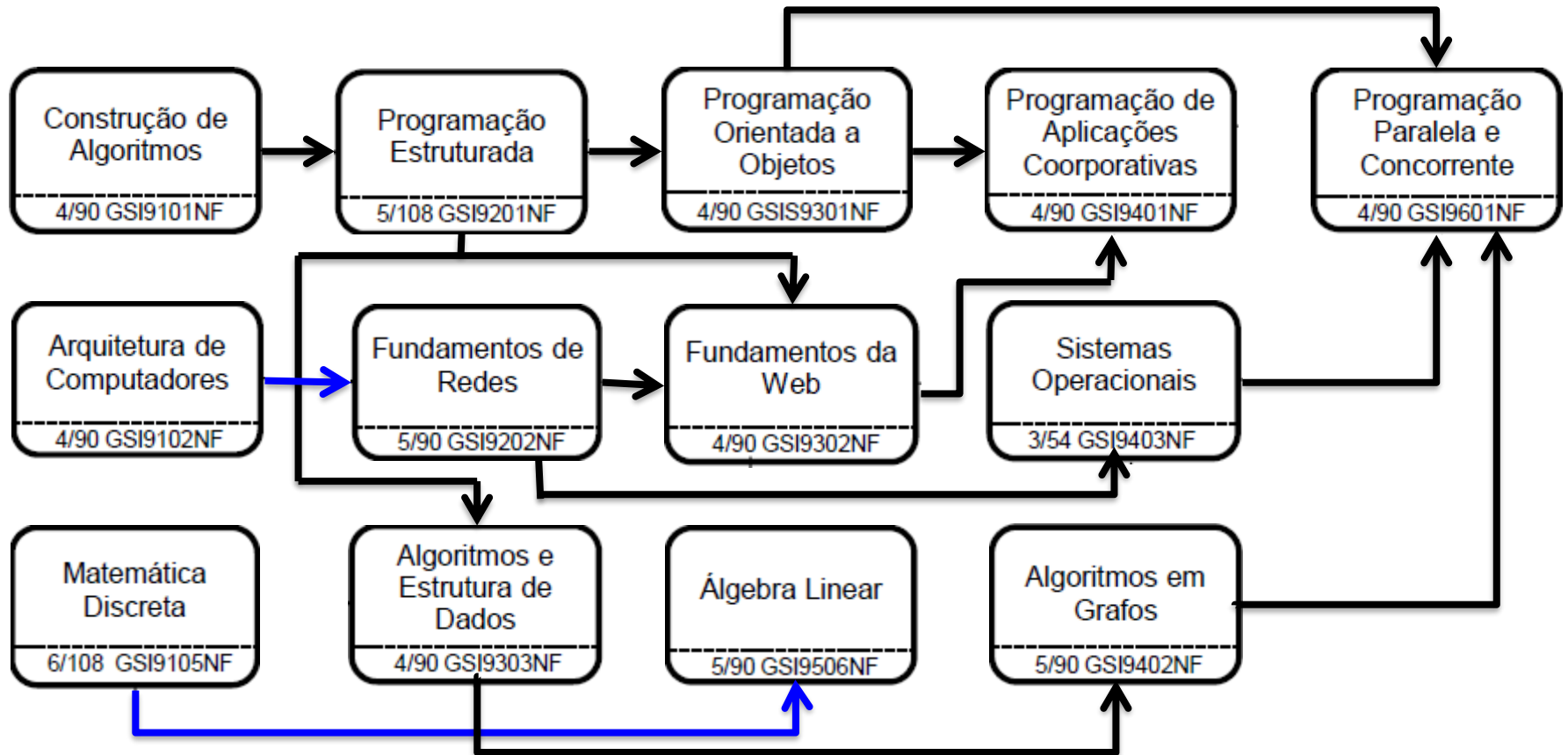
- Exemplo:



Ordenação topológica:

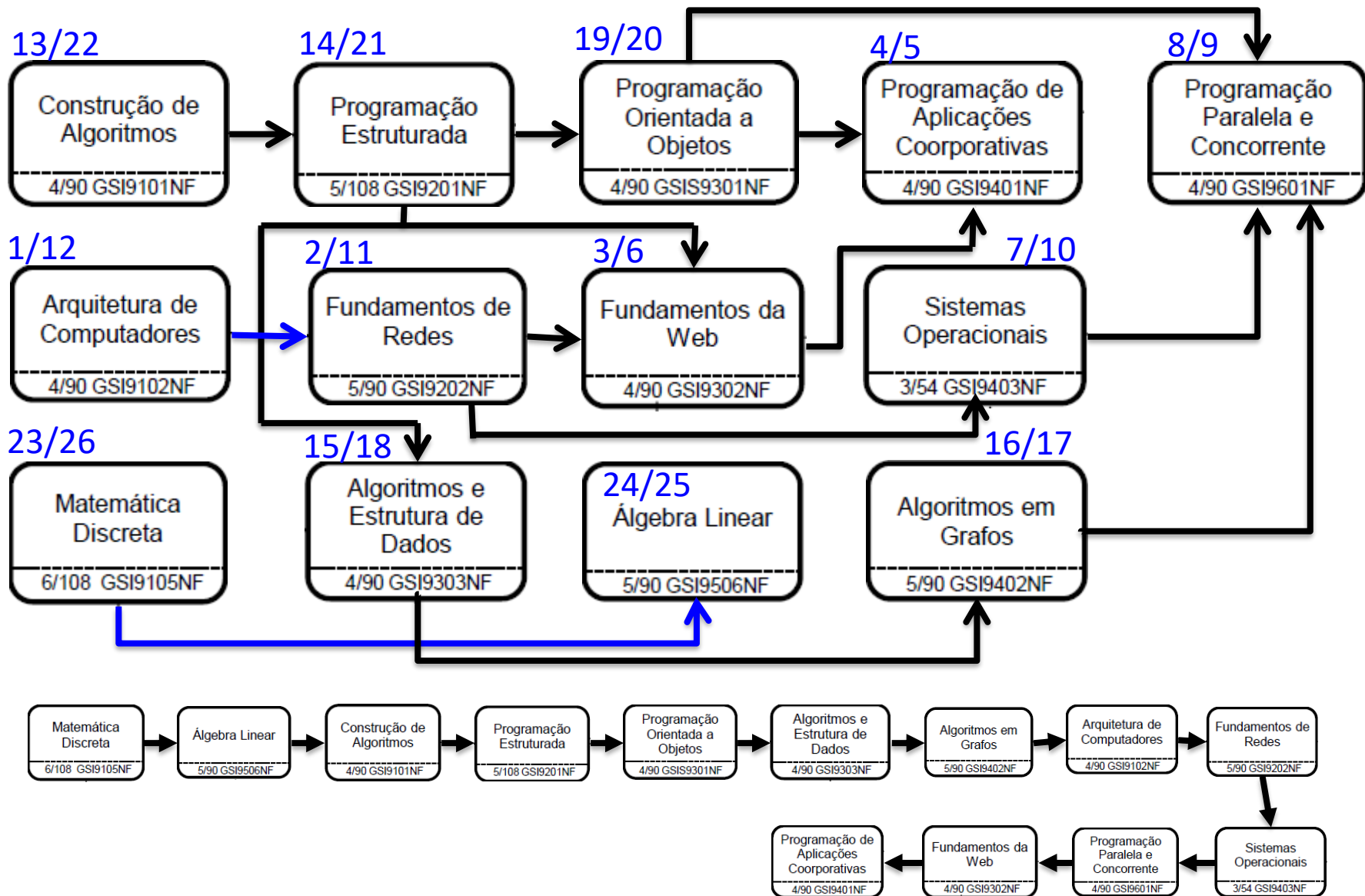


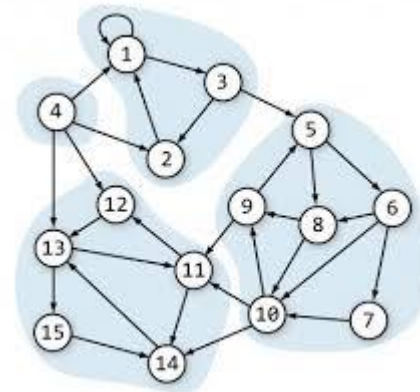
• Exercício 1



- Aplique a ordenação topológica para identificar uma sequência possível de matérias a serem cursadas.

Exercício 1

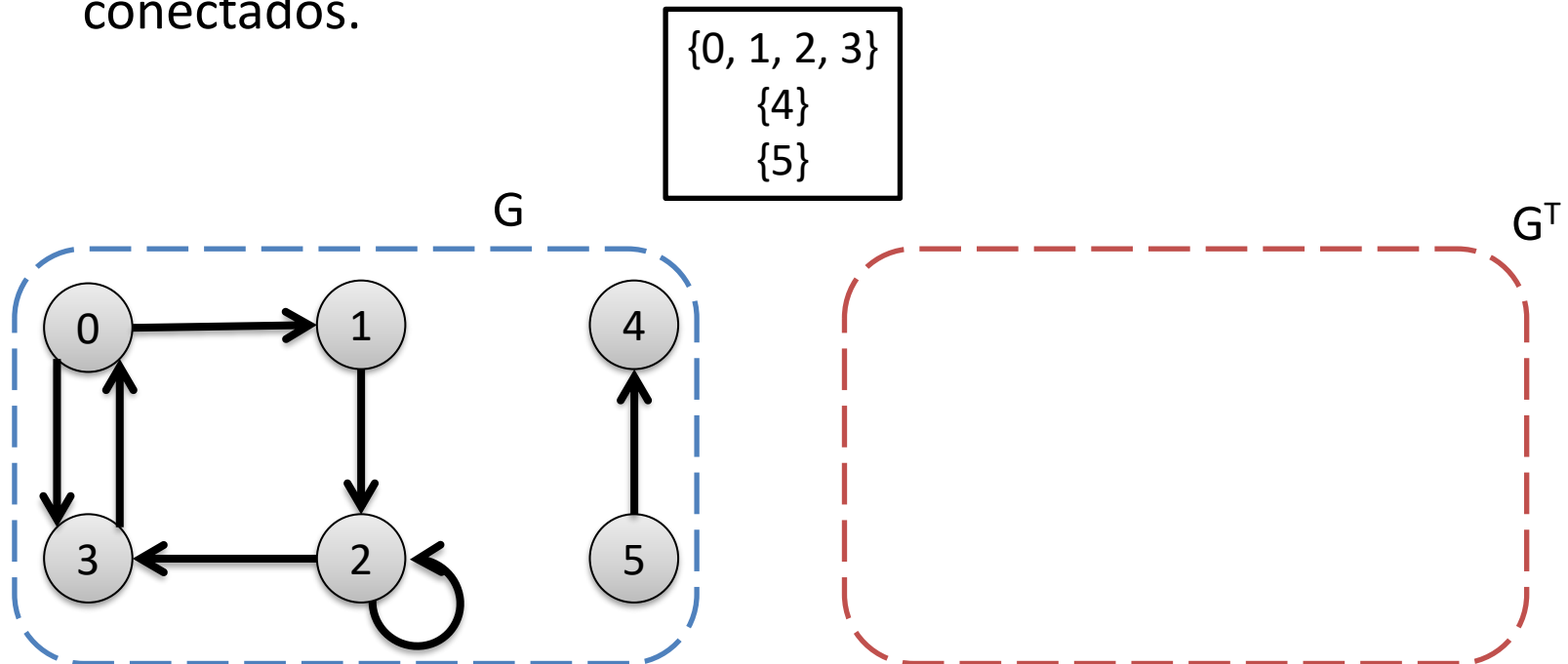




Algoritmos em grafos

- Componentes fortemente conectados

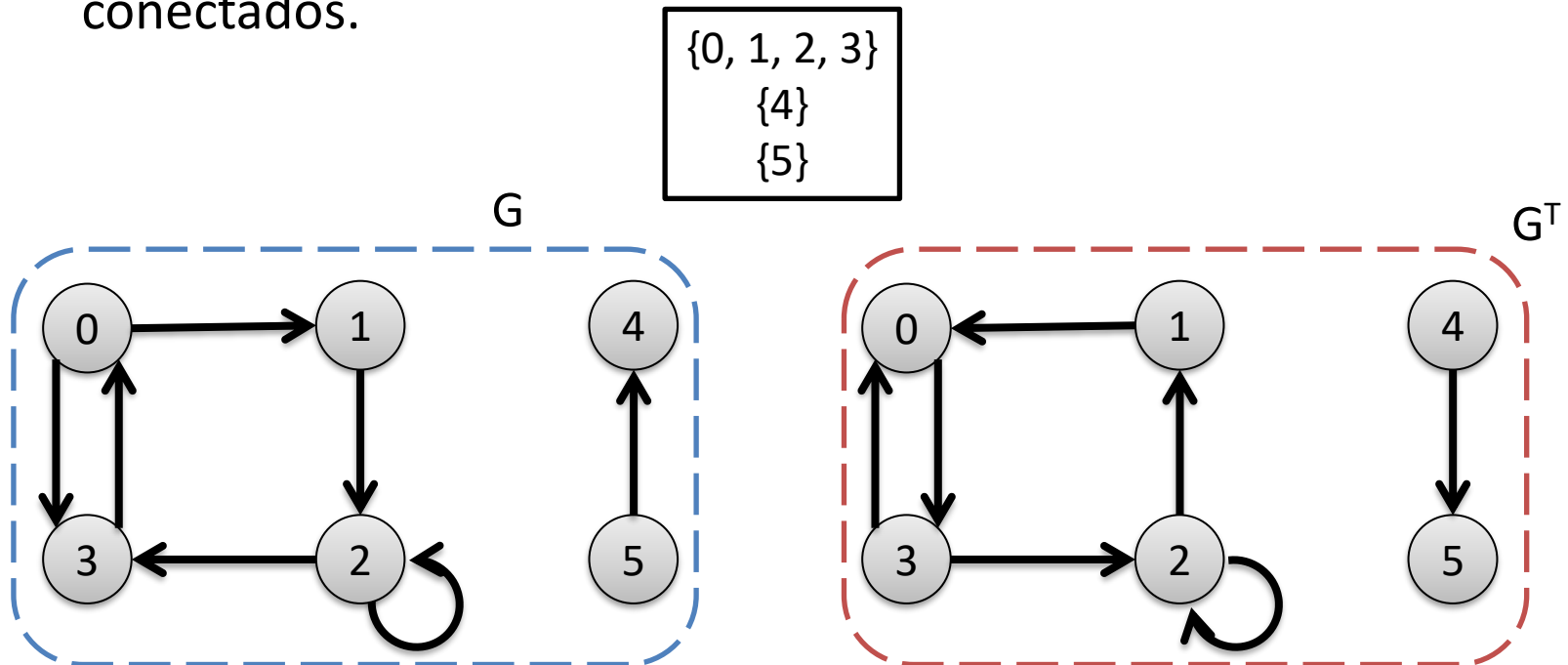
- **Propriedade:** G e G^T possuem os **mesmos** componentes fortemente conectados.



Algoritmos em grafos

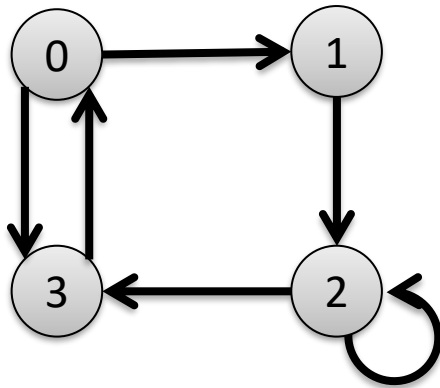
- Componentes fortemente conectados

- **Propriedade:** G e G^T possuem os **mesmos** componentes fortemente conectados.



Algoritmos em grafos

- **Componentes fortemente conectados**
 - **Componentes fortemente conectados** de um grafo direcionado são conjuntos de vértices sob a relação de “**mutuamente alcançáveis**”.



Componentes fortemente conexos:

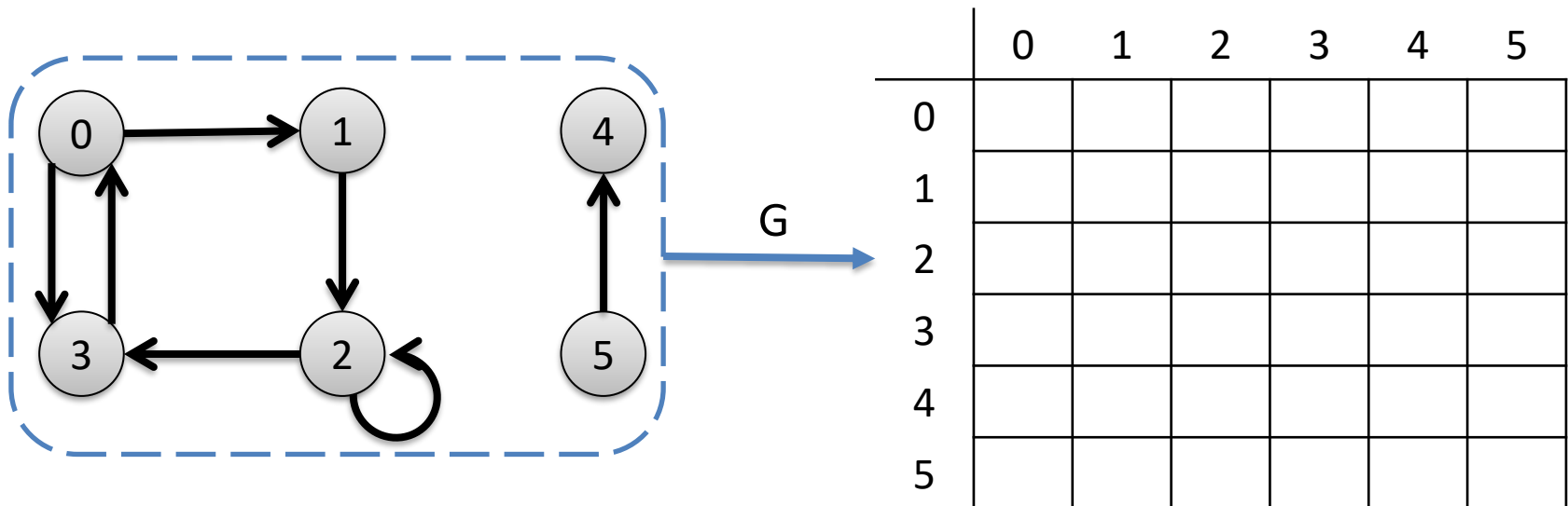
$\{0, 1, 2, 3\}$

$\{4\}$

$\{5\}$

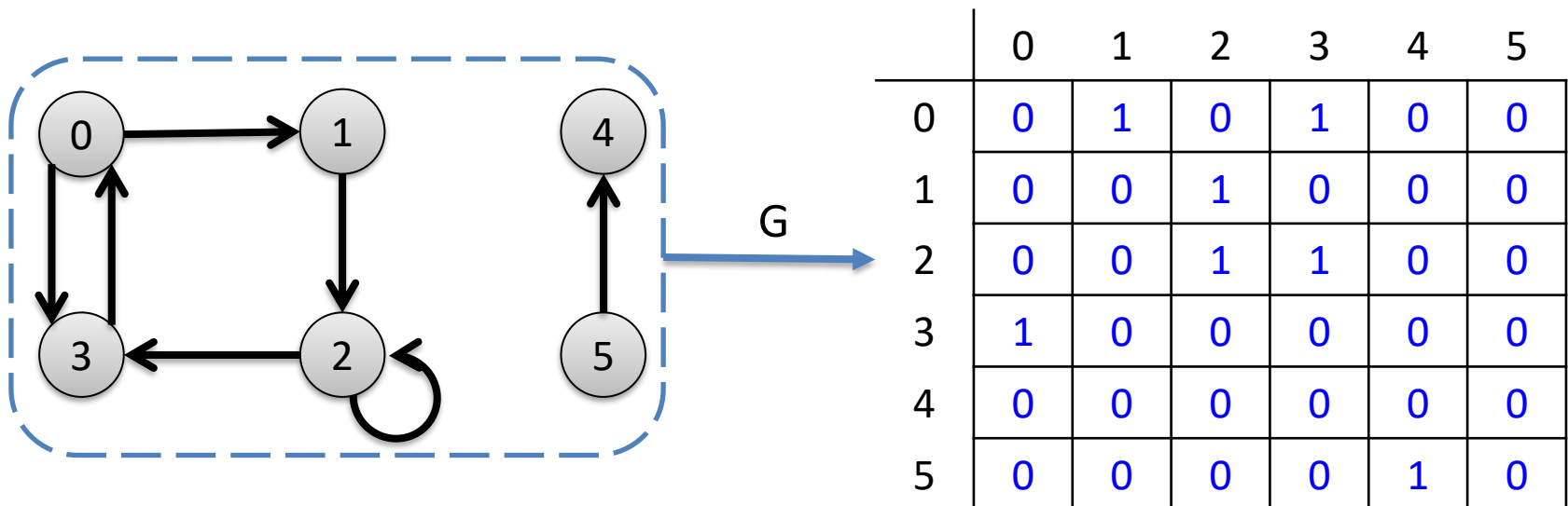
Algoritmos em grafos

- Componentes fortemente conectados
 - **Estratégia:** Utilizar duas **buscas em profundidade**. Uma no **grafo** e outra na **transposta do grafo**. Todo **subgrafo** encontrado no final da **segunda aplicação** corresponde a um **componente fortemente conectado**!



Algoritmos em grafos

- Componentes fortemente conectados
 - **Estratégia:** Utilizar duas **buscas em profundidade**. Uma no **grafo** e outra na **transposta do grafo**. Todo **subgrafo** encontrado no final da **segunda aplicação** corresponde a um **componente fortemente conectado**!



Algoritmos em grafos

- Componentes fortemente conectados
 - **Estratégia:** Utilizar duas **buscas em profundidade**. Uma no **grafo** e outra na **transposta do grafo**. Todo **subgrafo** encontrado no final da **segunda aplicação** corresponde a um **componente fortemente conectado**!

	0	1	2	3	4	5
0	0	1	0	1	0	0
1	0	0	1	0	0	0
2	0	0	1	1	0	0
3	1	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	1	0

G^T →

	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Algoritmos em grafos

- Componentes fortemente conectados
 - **Estratégia:** Utilizar duas **buscas em profundidade**. Uma no **grafo** e outra na **transposta do grafo**. Todo **subgrafo** encontrado no final da **segunda aplicação** corresponde a um **componente fortemente conectado**!

	0	1	2	3	4	5
0	0	1	0	1	0	0
1	0	0	1	0	0	0
2	0	0	1	1	0	0
3	1	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	1	0


G^T

	0	1	2	3	4	5
0	0	0	0	1	0	0
1	1	0	0	0	0	0
2	0	1	1	0	0	0
3	1	0	1	0	0	0
4	0	0	0	0	0	1
5	0	0	0	0	0	0

Algoritmos em grafos

- **Componentes fortemente conectados**
 - **Estratégia:** Utilizar duas **buscas em profundidade**. Uma no **grafo** e outra na **transposta do grafo**. Todo **subgrafo** encontrado no final da **segunda aplicação** corresponde a um **componente fortemente conectado**!

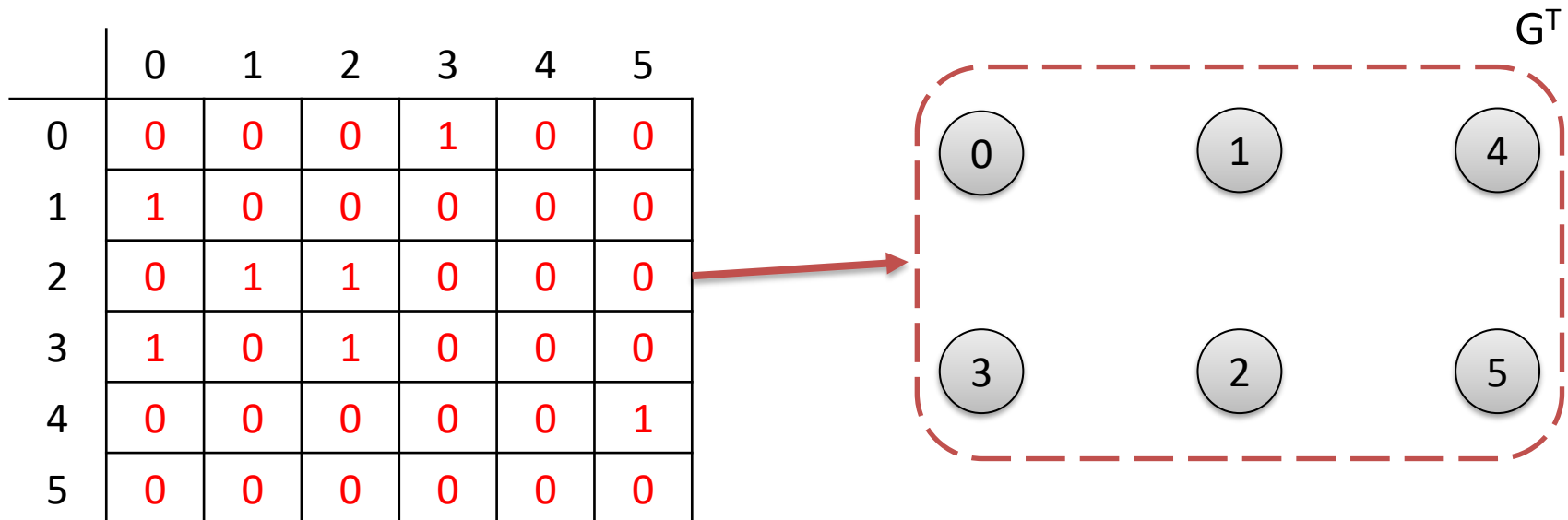
	0	1	2	3	4	5
0	0	0	0	1	0	0
1	1	0	0	0	0	0
2	0	1	1	0	0	0
3	1	0	1	0	0	0
4	0	0	0	0	0	1
5	0	0	0	0	0	0



The diagram shows a large, empty rounded rectangle with a dashed red border, representing the transpose graph G^T . A red arrow points from the right side of the adjacency matrix table to the left side of this rectangle. The label G^T is positioned at the top right of the rectangle.

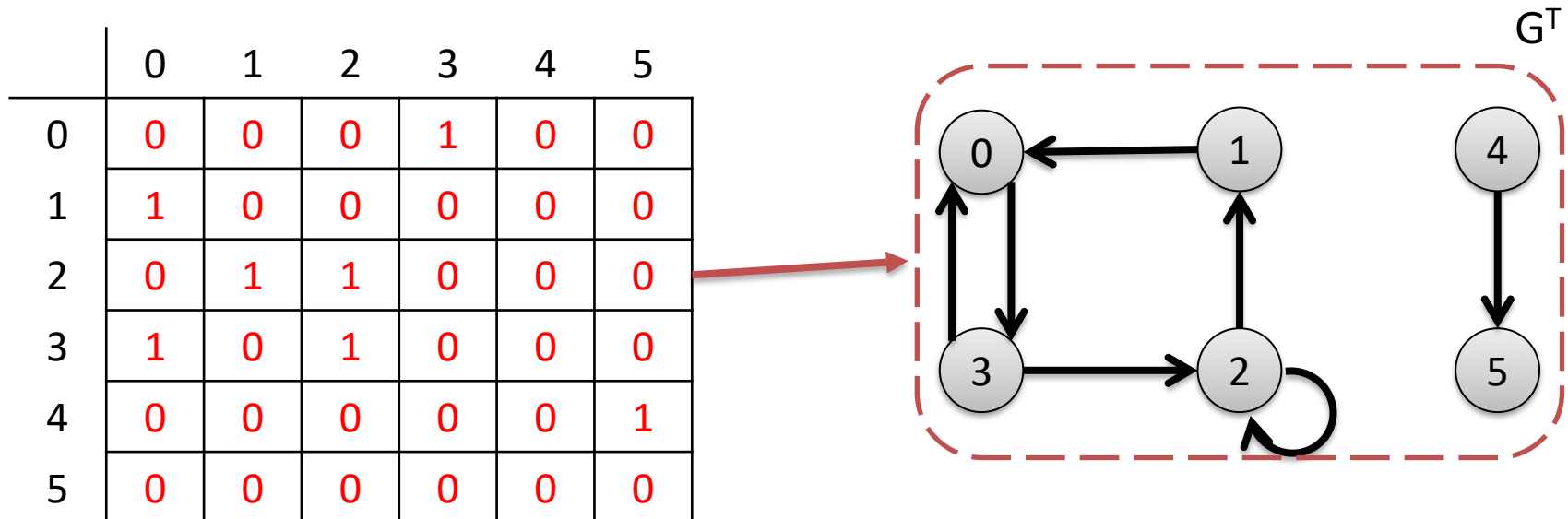
Algoritmos em grafos

- Componentes fortemente conectados
 - **Estratégia:** Utilizar duas **buscas em profundidade**. Uma no **grafo** e outra na **transposta do grafo**. Todo **subgrafo** encontrado no final da **segunda aplicação** corresponde a um **componente fortemente conectado**!



Algoritmos em grafos

- Componentes fortemente conectados
 - **Estratégia:** Utilizar duas **buscas em profundidade**. Uma no **grafo** e outra na **transposta do grafo**. Todo **subgrafo** encontrado no final da **segunda aplicação** corresponde a um **componente fortemente conectado**!



Algoritmos em grafos

- Componentes fortemente conectados

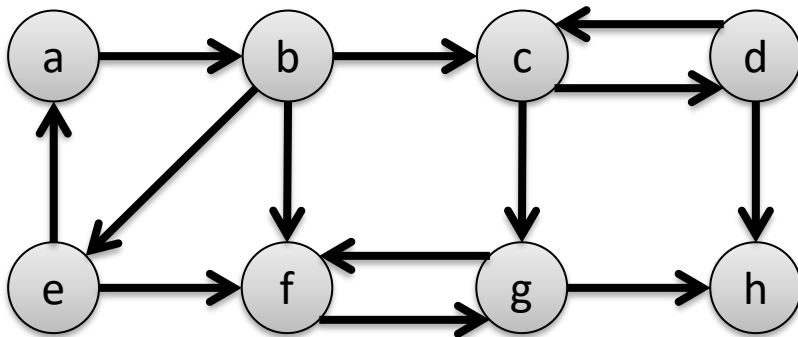
- Algoritmo:

1. Aplicar a **busca em profundidade** em G para obter os **tempos de término** para cada vértice.
2. Obtém G^T .
3. Aplicar a **busca em profundidade** em G^T , começando pelo nó de **maior tempo de término**, encontrado no passo 1.
4. Enquanto houver **vértices restantes**, iniciar uma **nova busca em profundidade em G^T** a partir do nó de **maior tempo de término** dentre os restantes.
5. Retorne os vértices de cada subgrafo obtido como um **componente fortemente conectado** separado.

Algoritmos em grafos

- **Componentes fortemente conectados**

– Exemplo:

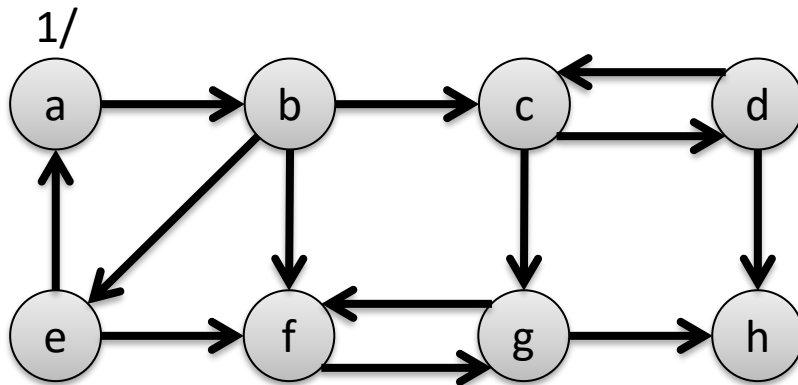


	a	b	c	d	e	f	g	h
f								

Algoritmos em grafos

- **Componentes fortemente conectados**

– Exemplo:

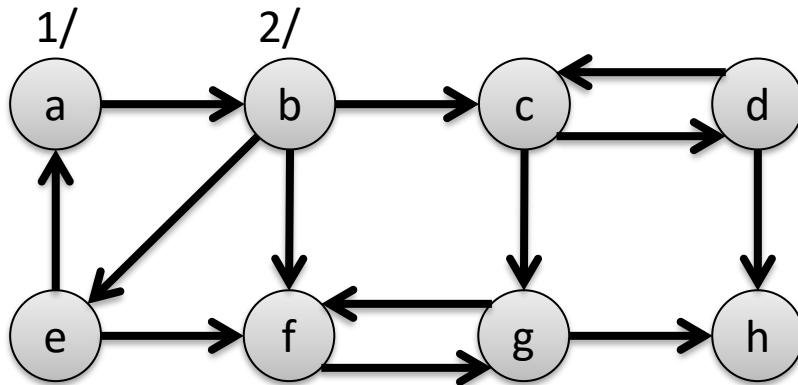


	a	b	c	d	e	f	g	h
f								

Algoritmos em grafos

- **Componentes fortemente conectados**

– Exemplo:

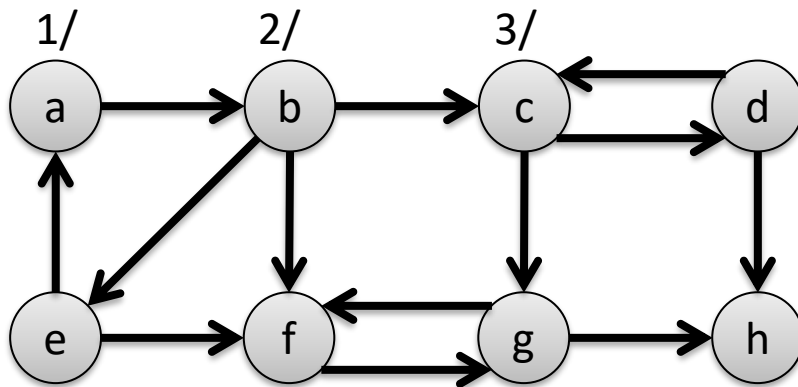


	a	b	c	d	e	f	g	h
f								

Algoritmos em grafos

- **Componentes fortemente conectados**

– Exemplo:

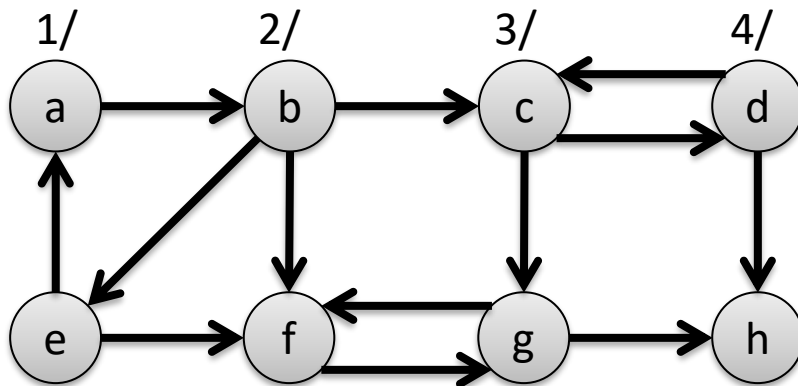


	a	b	c	d	e	f	g	h
f								

Algoritmos em grafos

- **Componentes fortemente conectados**

– Exemplo:

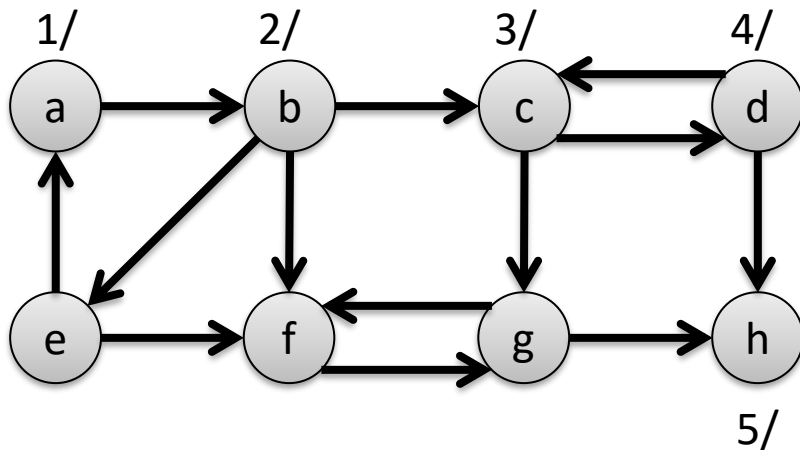


	a	b	c	d	e	f	g	h
f								

Algoritmos em grafos

- **Componentes fortemente conectados**

– Exemplo:

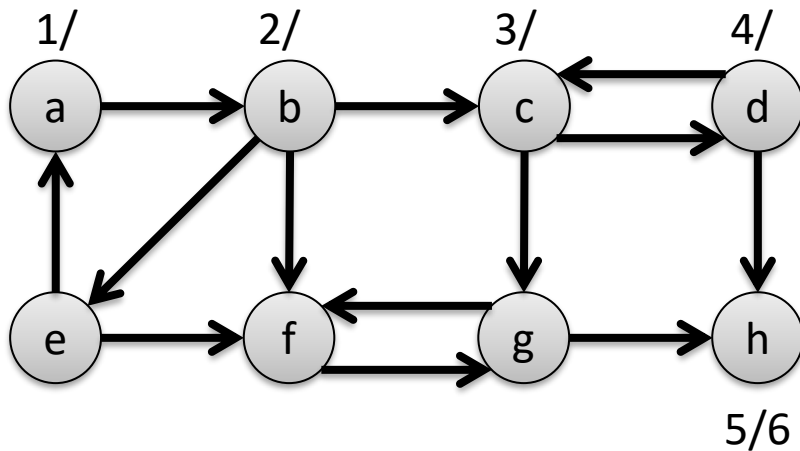


	a	b	c	d	e	f	g	h
f								

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

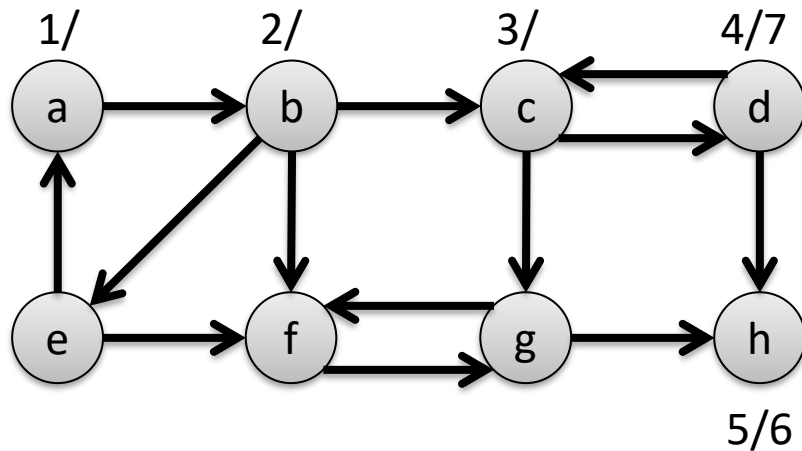


	a	b	c	d	e	f	g	h
f								6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

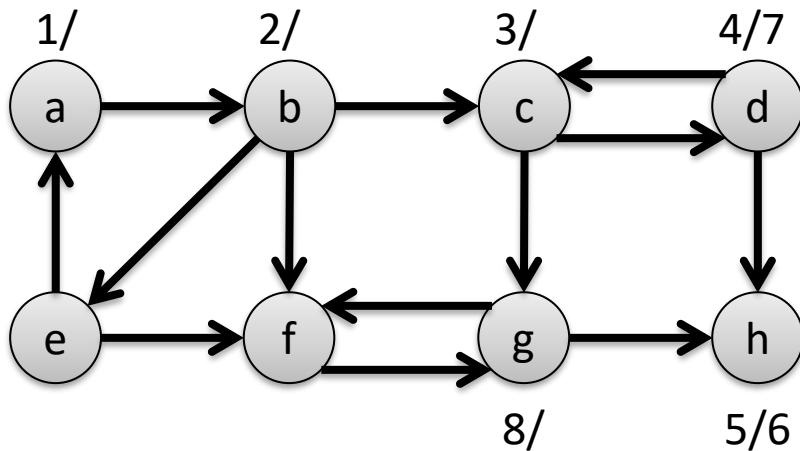


	a	b	c	d	e	f	g	h
f				7				6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

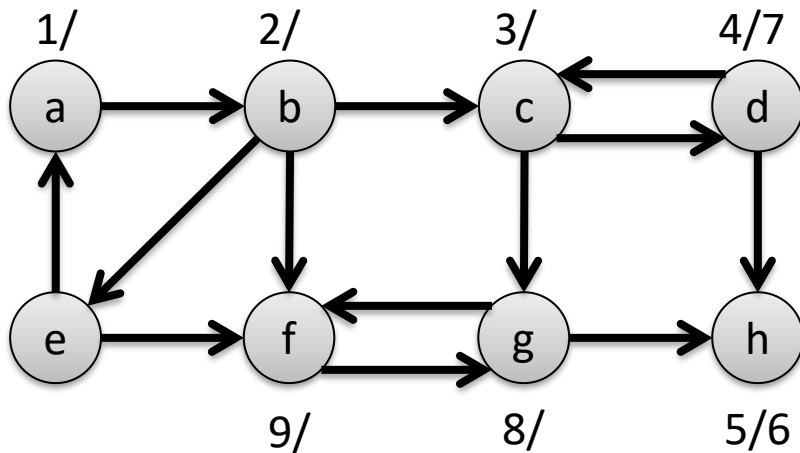


	a	b	c	d	e	f	g	h
f				7				6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

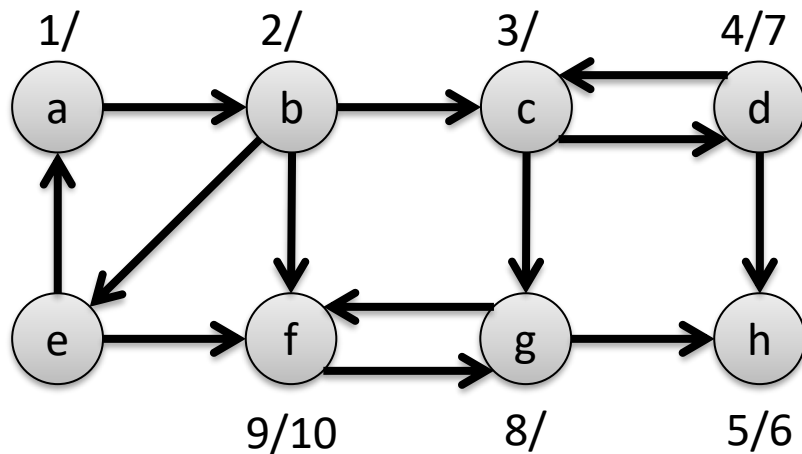


	a	b	c	d	e	f	g	h
f				7				6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

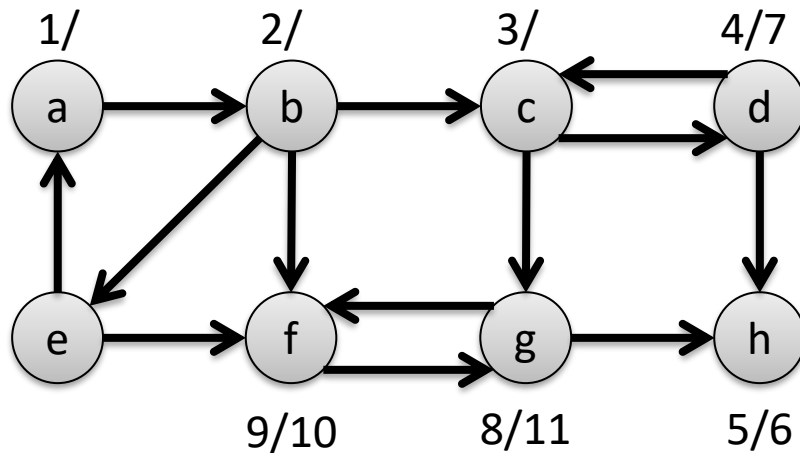


	a	b	c	d	e	f	g	h
f				7		10		6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

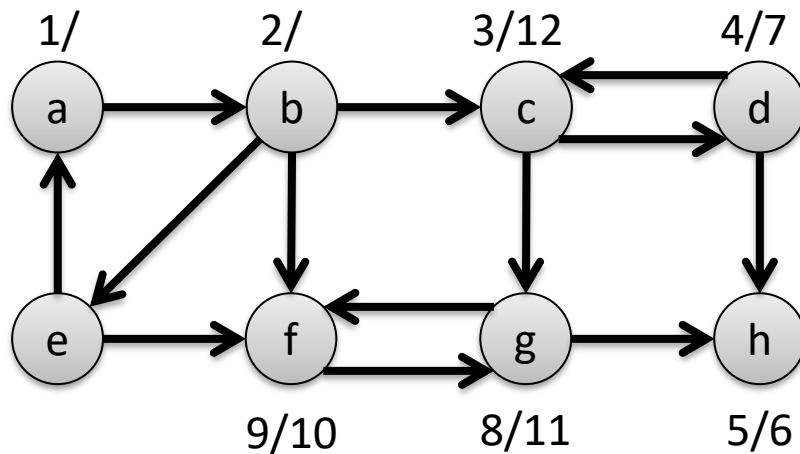


	a	b	c	d	e	f	g	h
f				7		10	11	6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

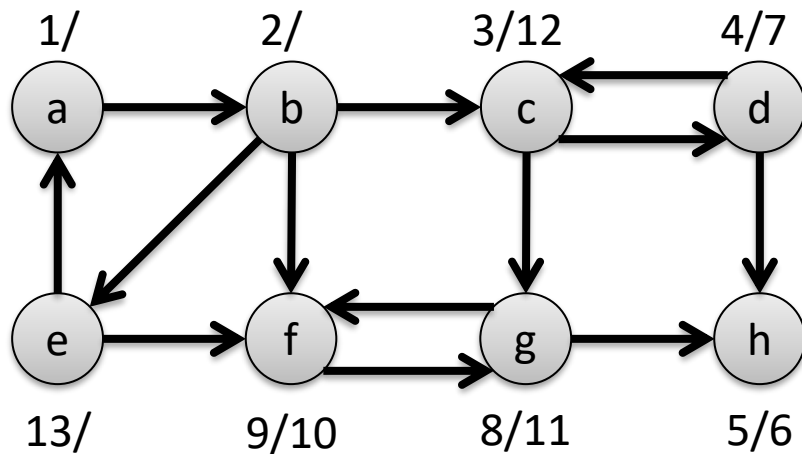


	a	b	c	d	e	f	g	h
f			12	7		10	11	6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

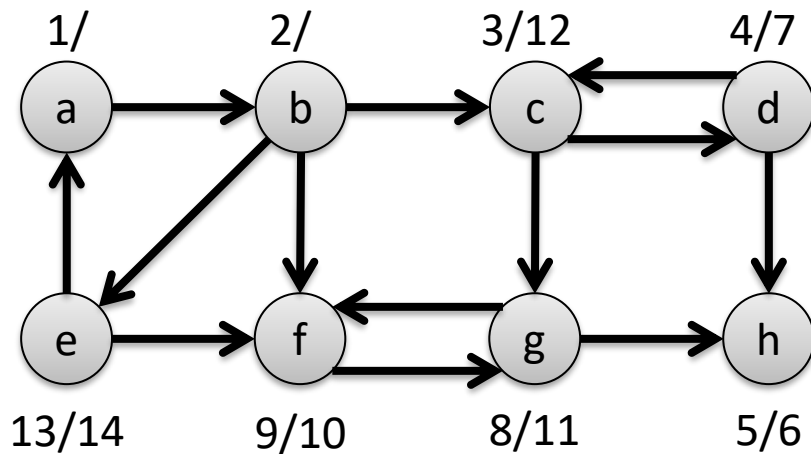


	a	b	c	d	e	f	g	h
f			12	7		10	11	6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

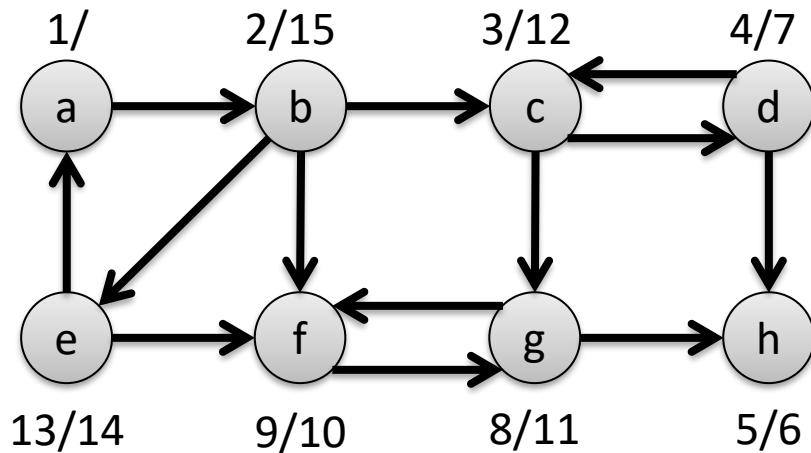


	a	b	c	d	e	f	g	h
f			12	7	14	10	11	6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

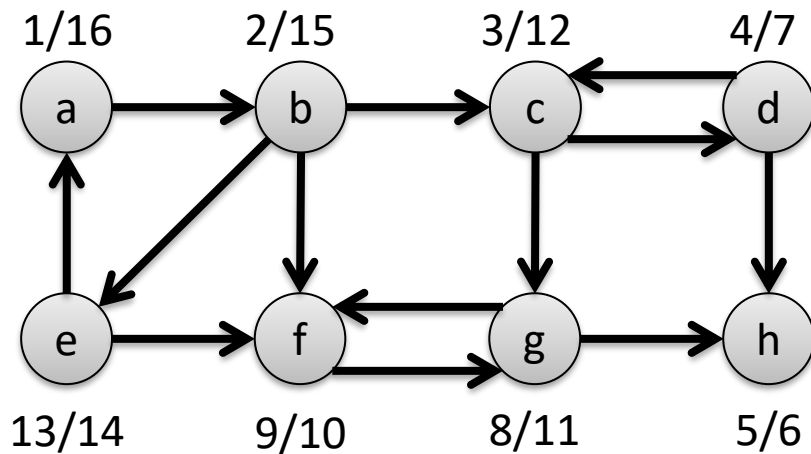


	a	b	c	d	e	f	g	h
f		15	12	7	14	10	11	6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

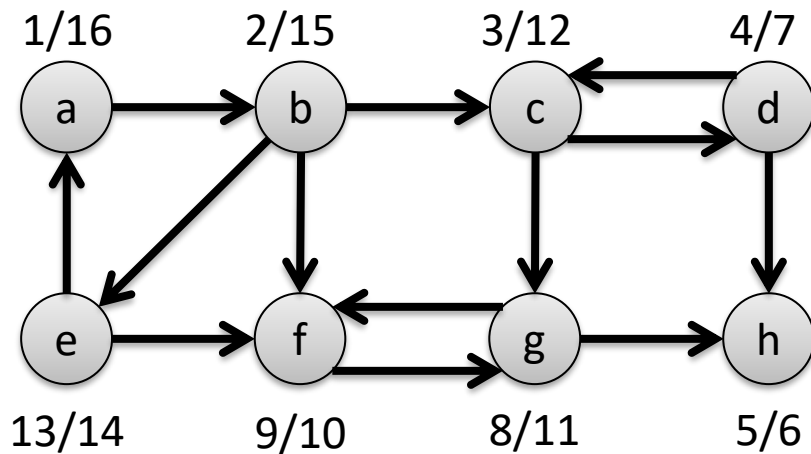


	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6

Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:

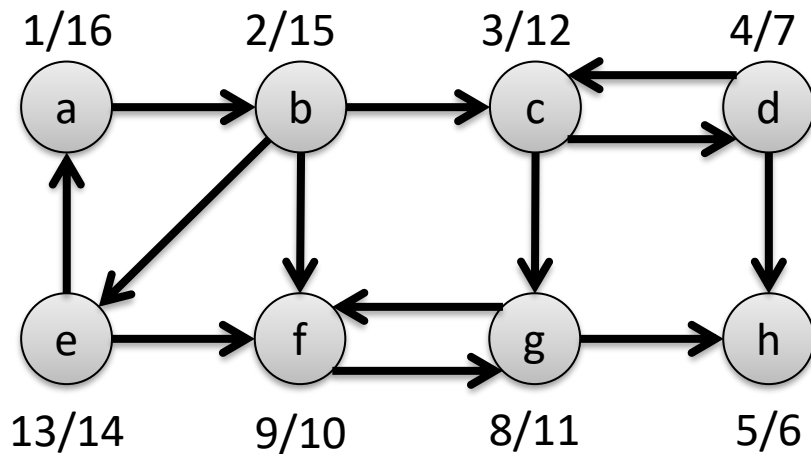


	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6

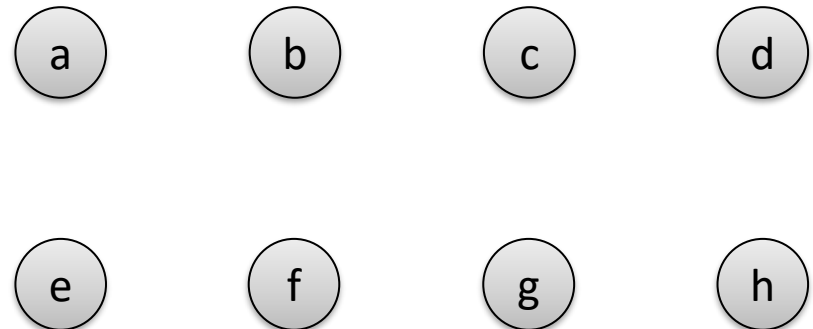
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



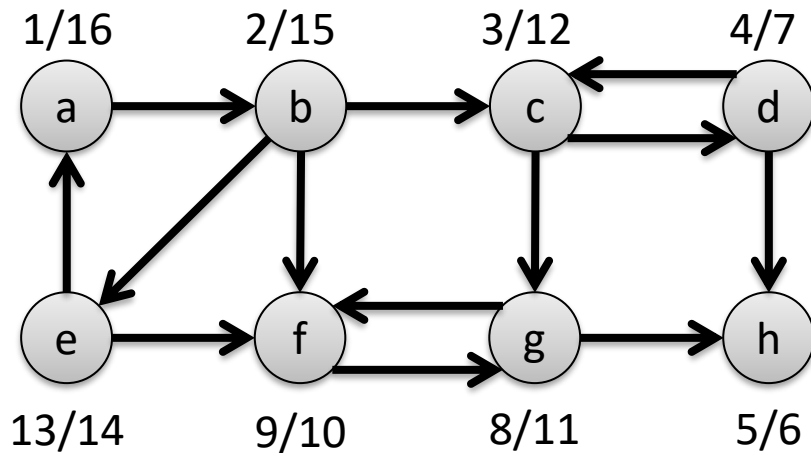
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



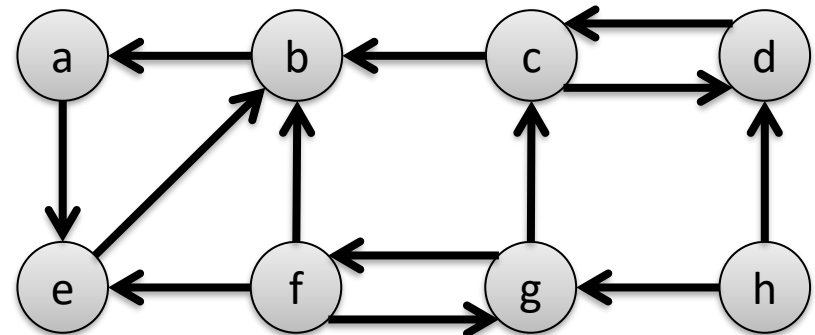
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



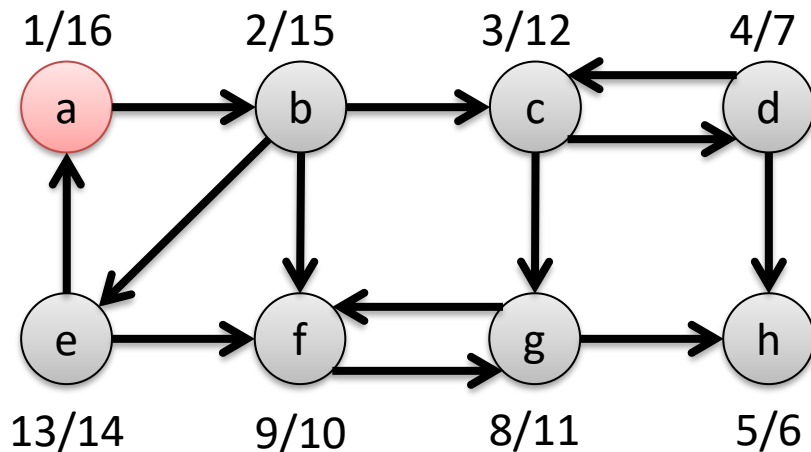
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



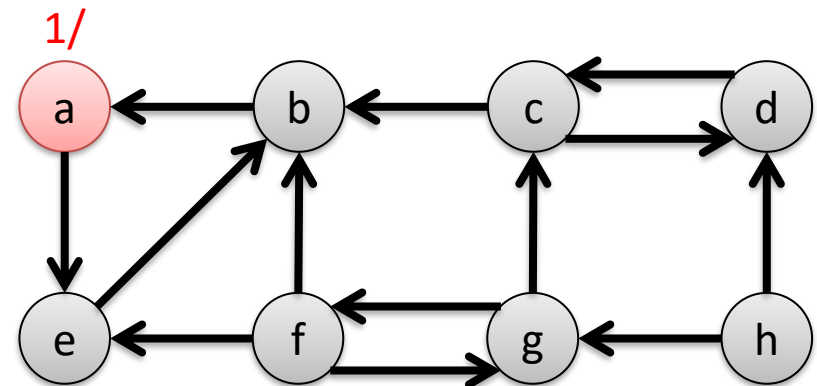
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



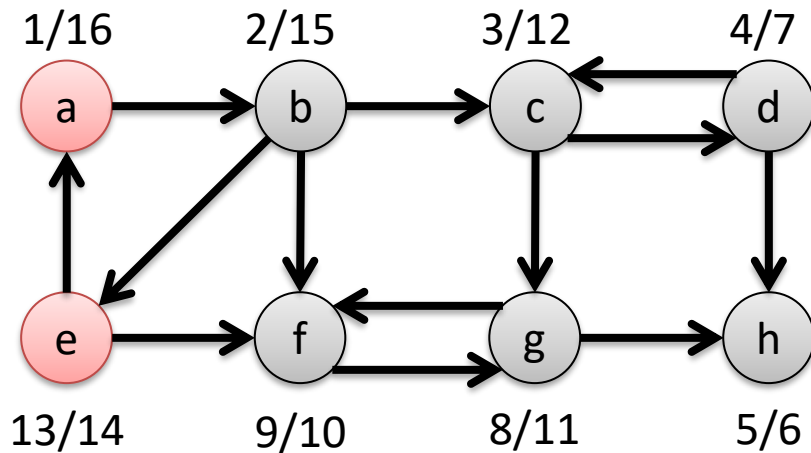
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



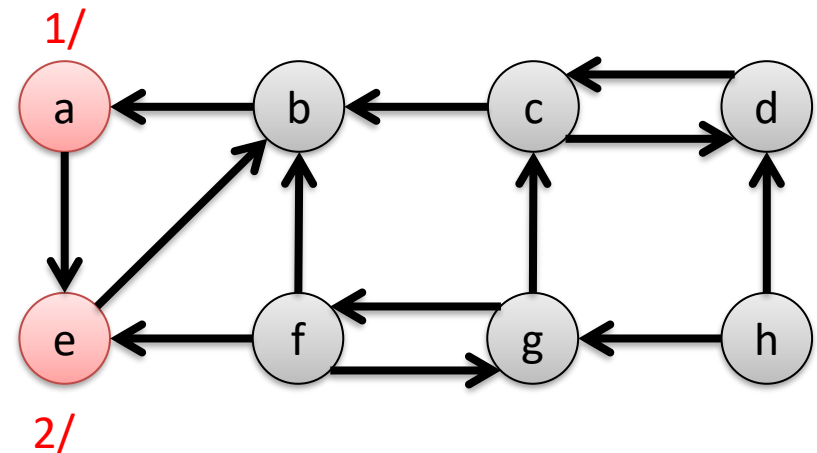
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



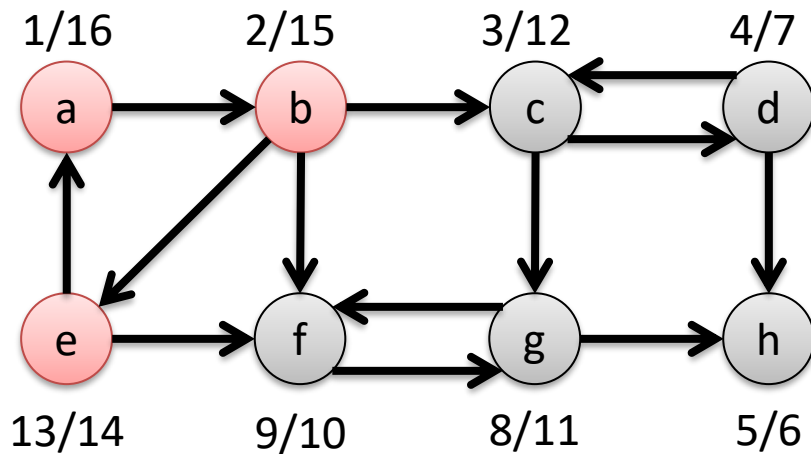
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



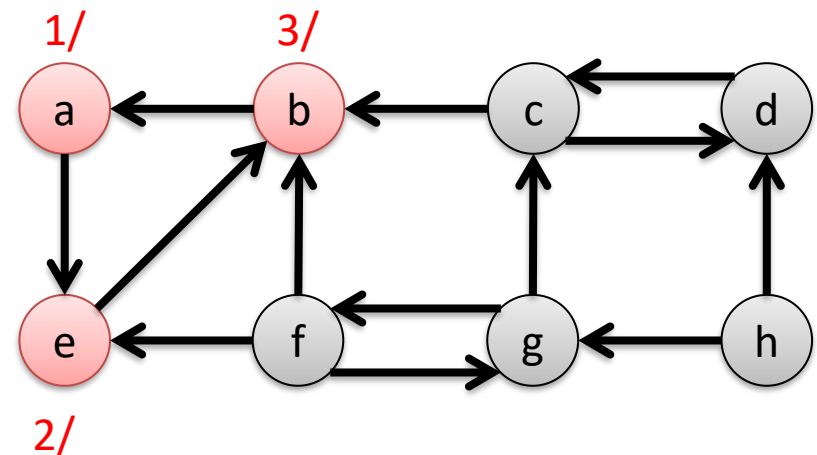
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



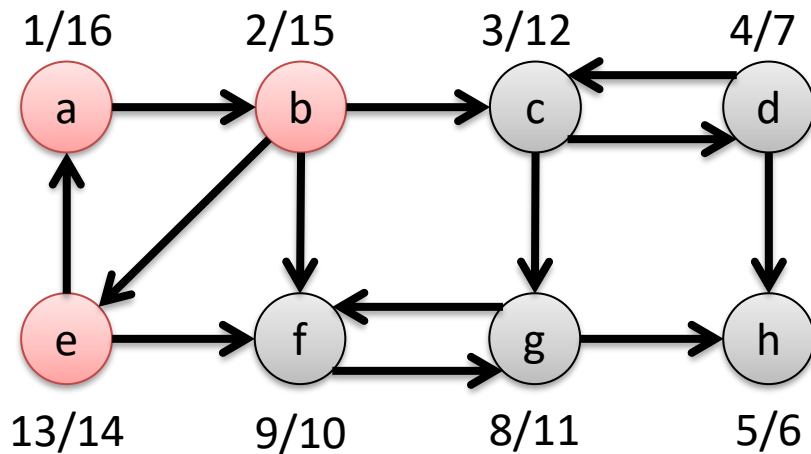
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



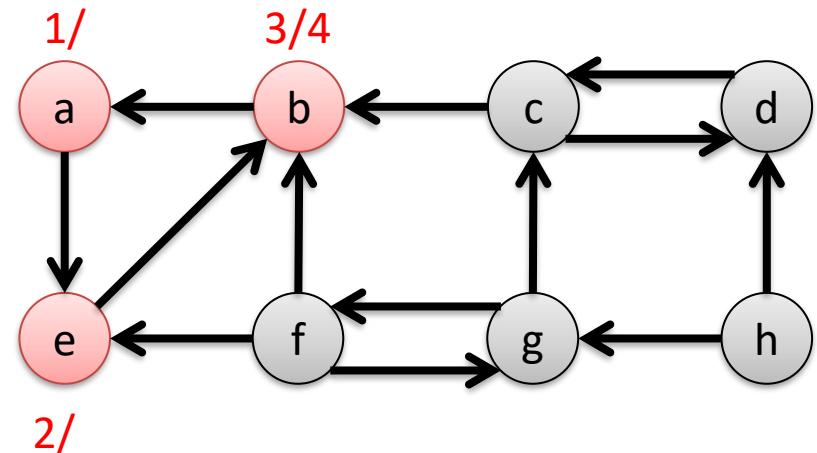
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



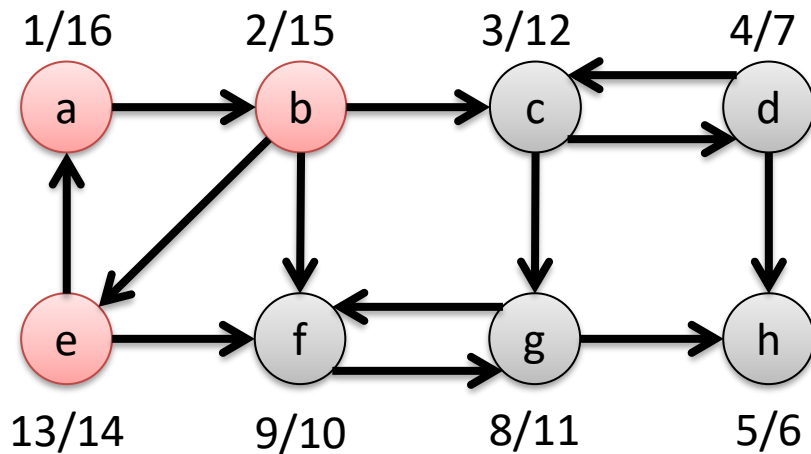
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



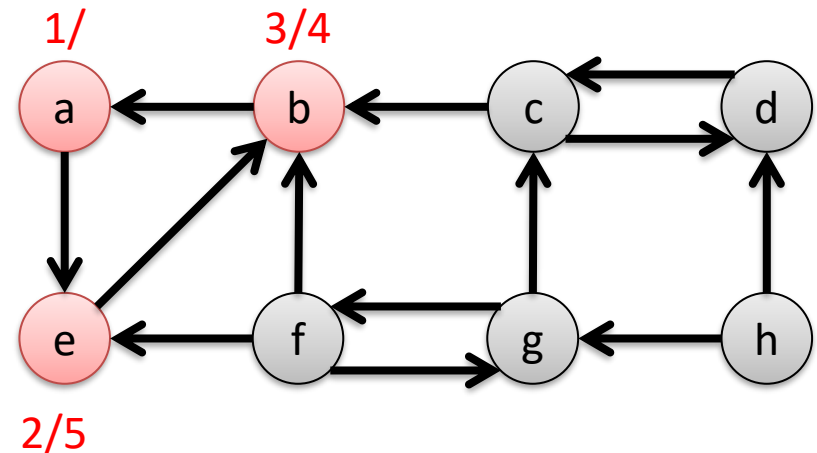
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



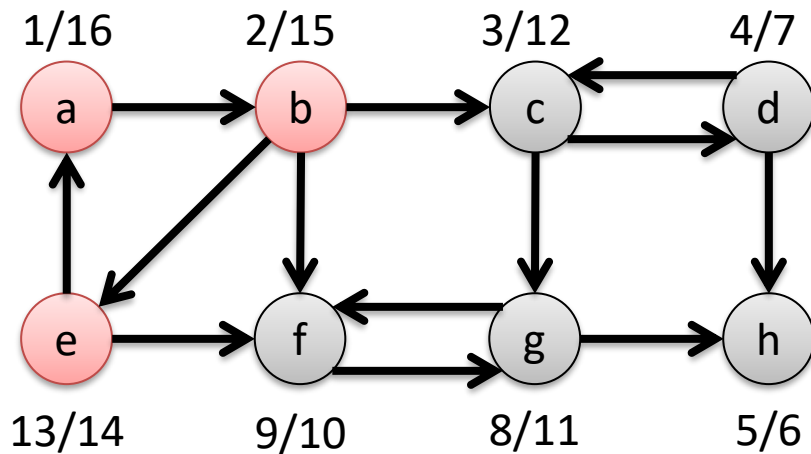
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



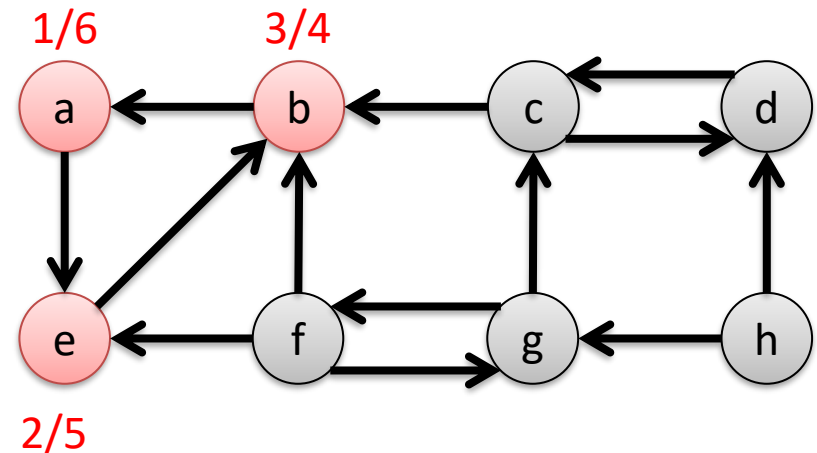
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



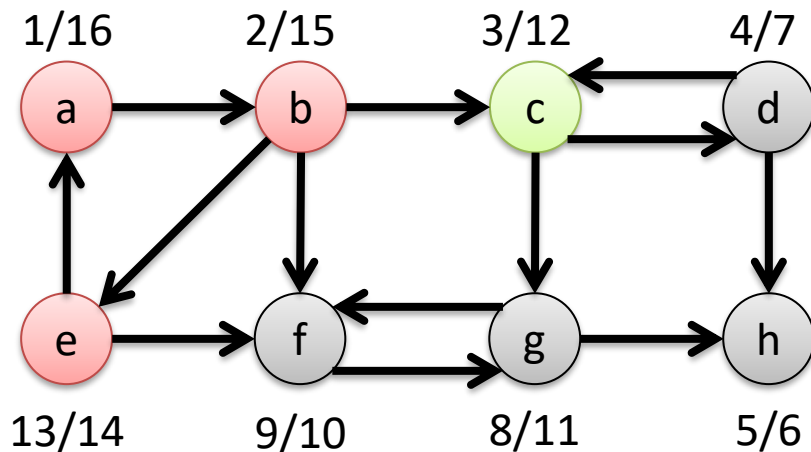
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



Algoritmos em grafos

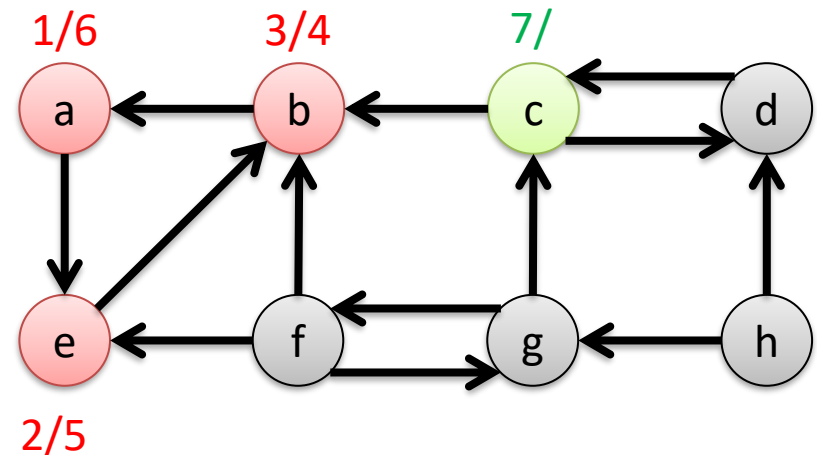
- Componentes fortemente conectados

– Exemplo:



	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6

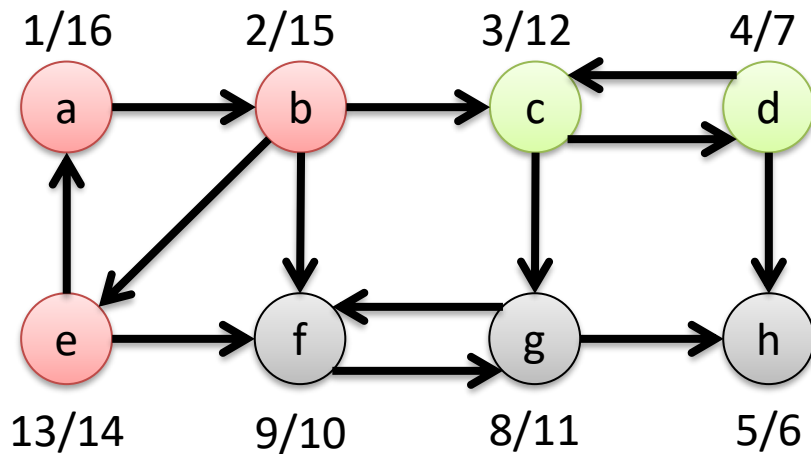
↑ ↑



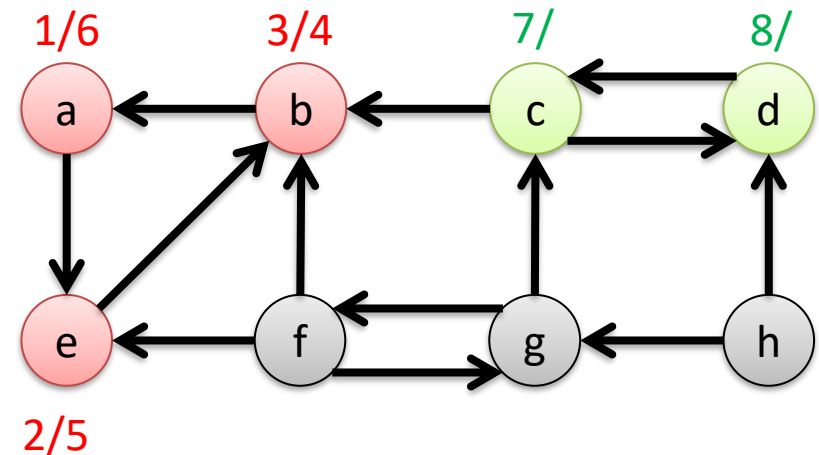
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



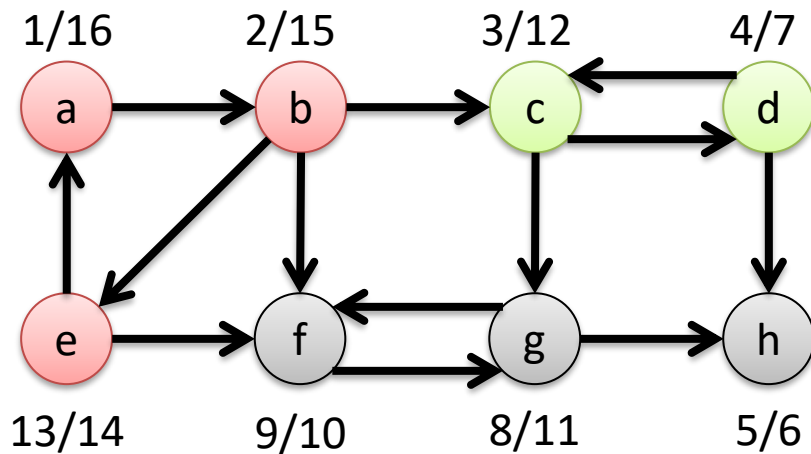
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



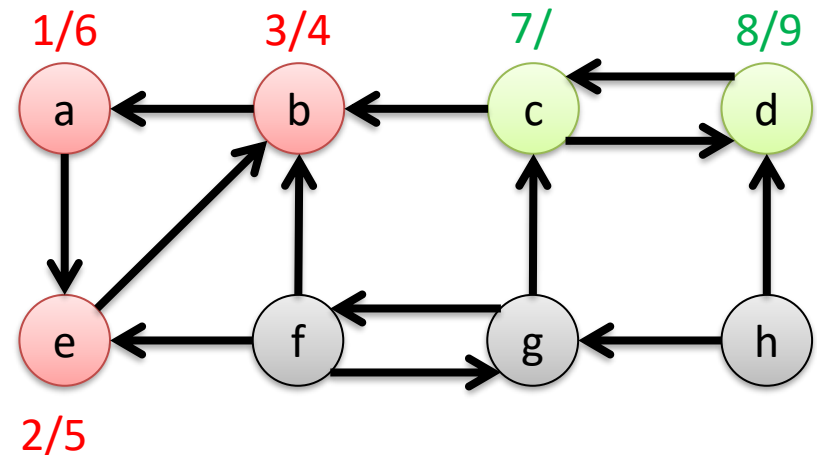
Algoritmos em grafos

- Componentes fortemente conectados

– Exemplo:



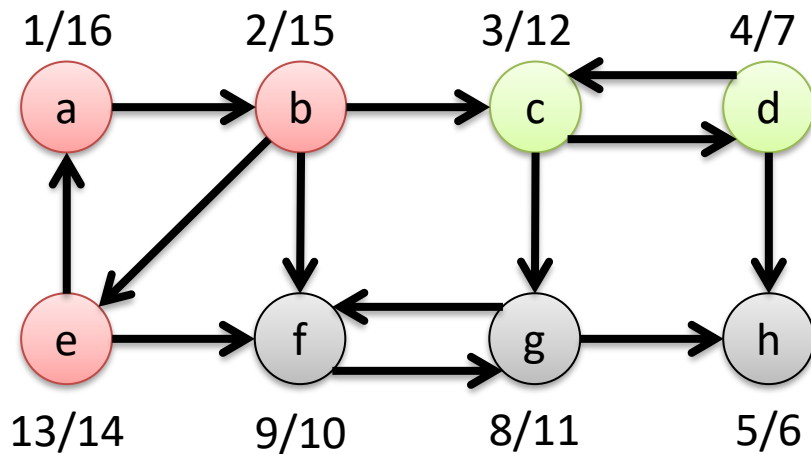
	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6



Algoritmos em grafos

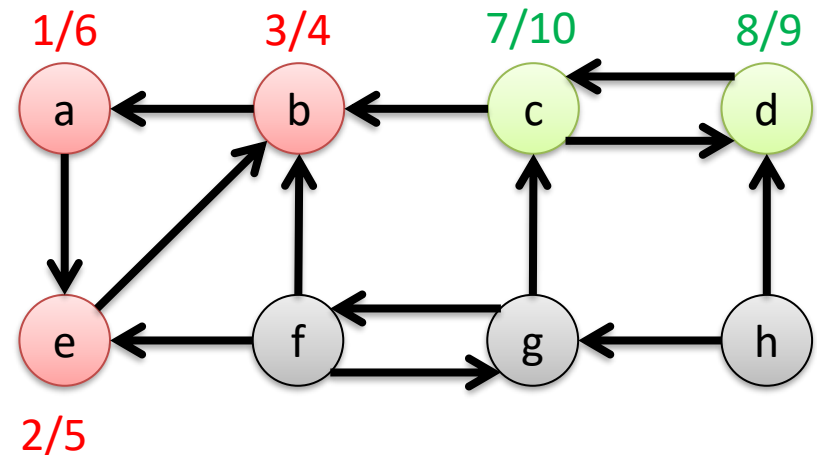
- Componentes fortemente conectados**

– Exemplo:



	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6

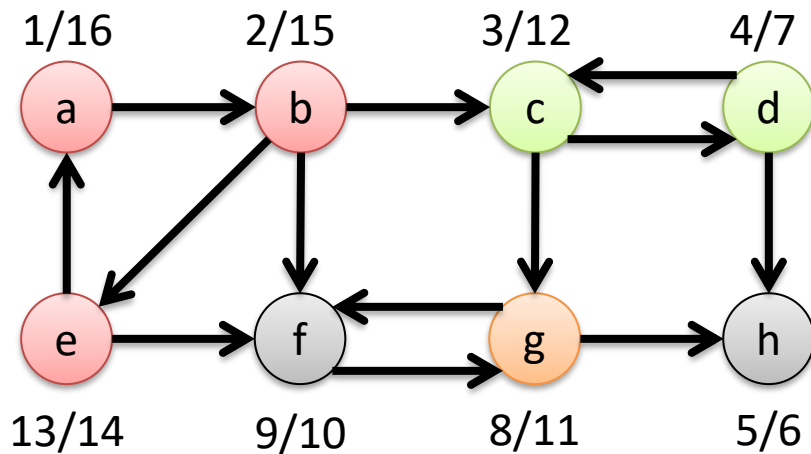
↑ (red arrow under 16) ↑ (green arrow under 12)



Algoritmos em grafos

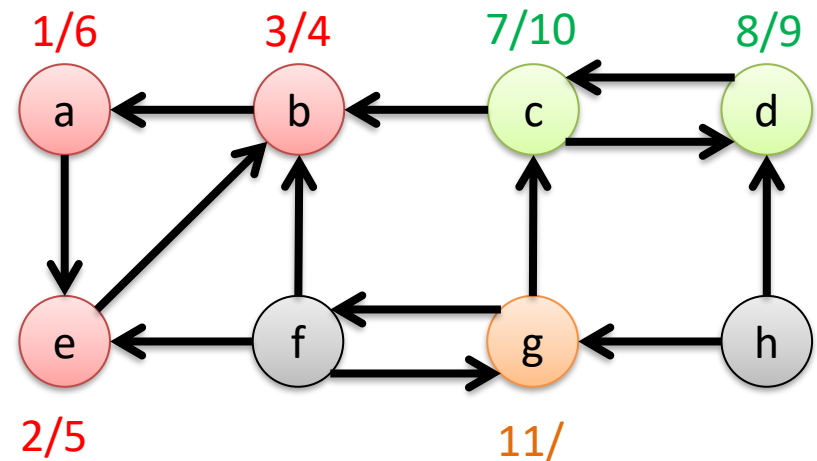
- Componentes fortemente conectados

– Exemplo:



	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6

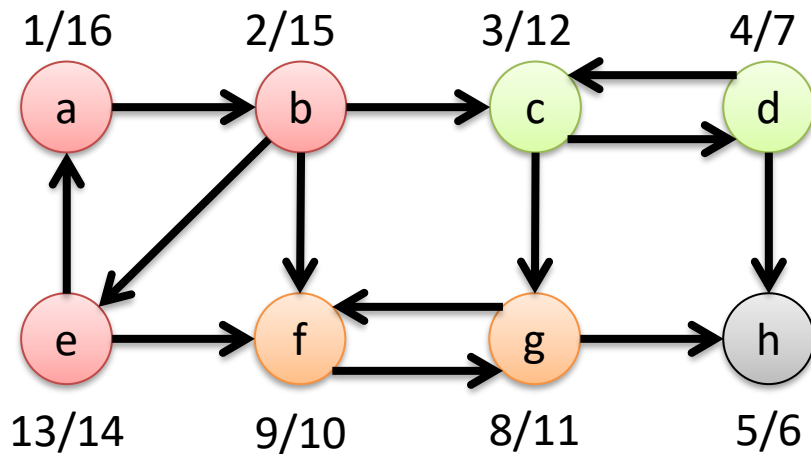
Arrows point from the table to the nodes: red arrow from 16 to a, green arrow from 12 to c, orange arrow from 11 to g.



Algoritmos em grafos

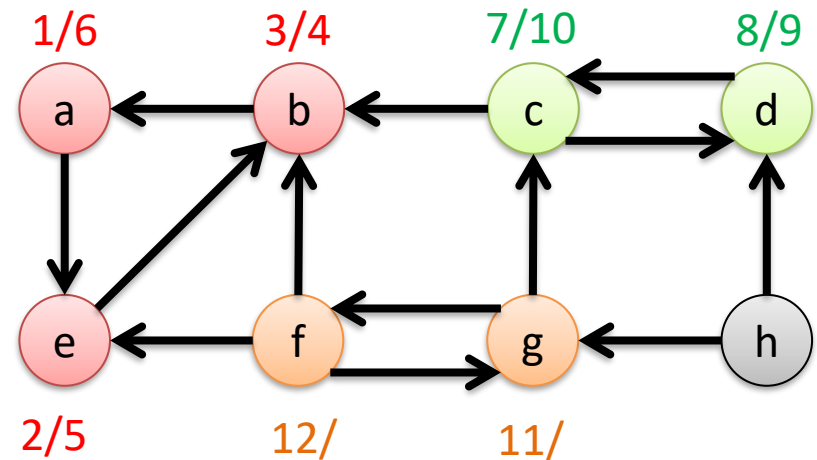
- Componentes fortemente conectados**

– Exemplo:



	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6

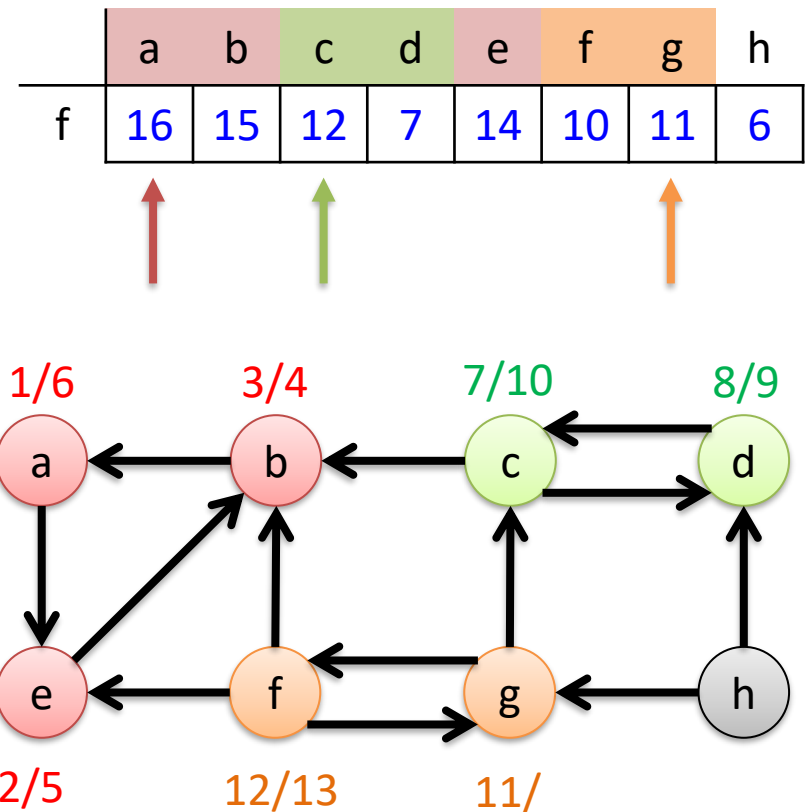
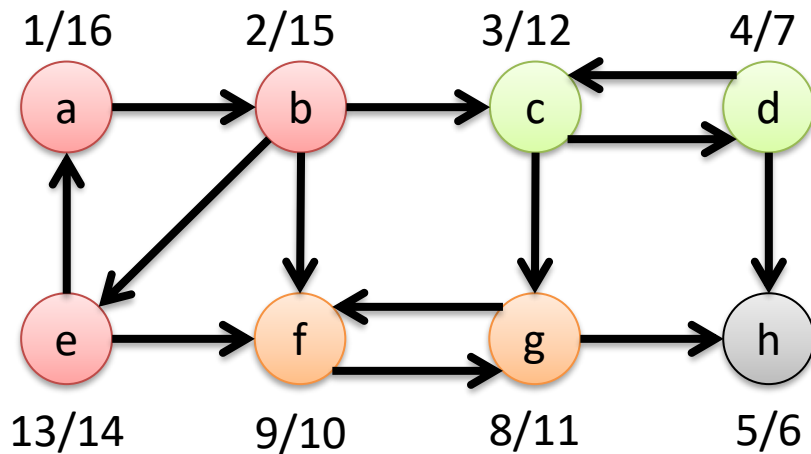
Arrows pointing up to the table: red arrow under 'a', green arrow under 'c', orange arrow under 'g'.



Algoritmos em grafos

- Componentes fortemente conectados

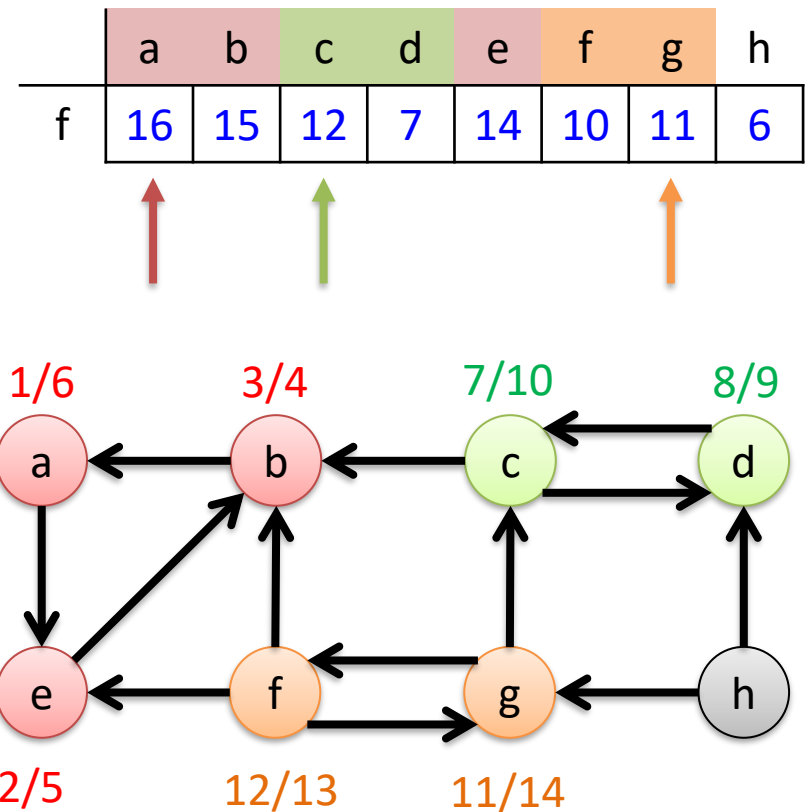
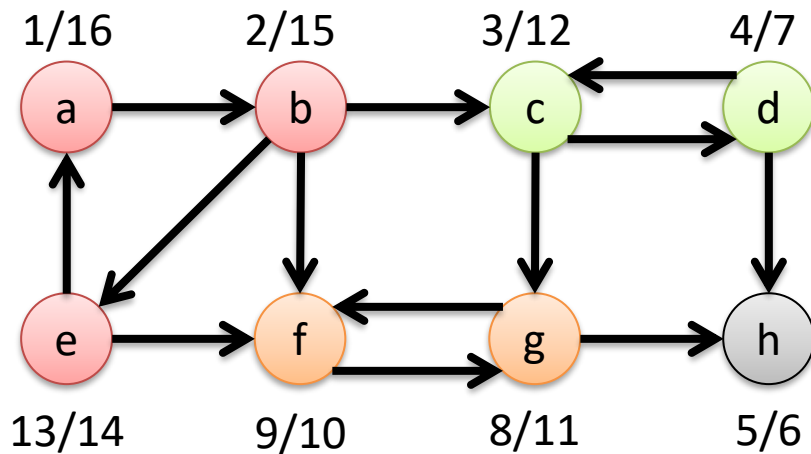
– Exemplo:



Algoritmos em grafos

- Componentes fortemente conectados

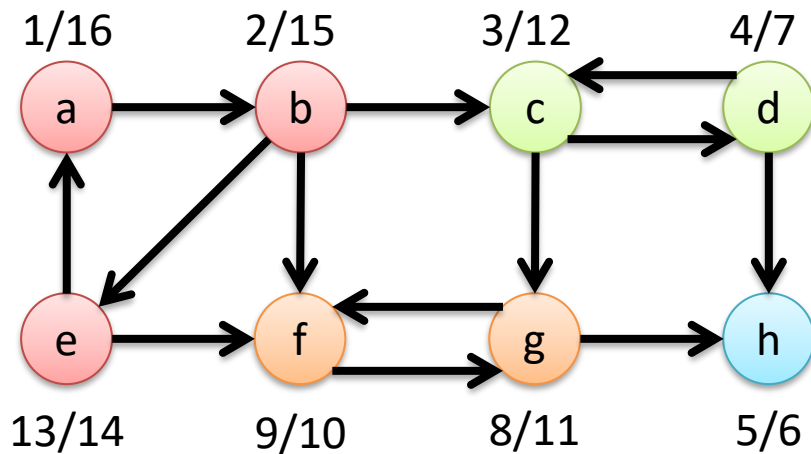
– Exemplo:



Algoritmos em grafos

- Componentes fortemente conectados**

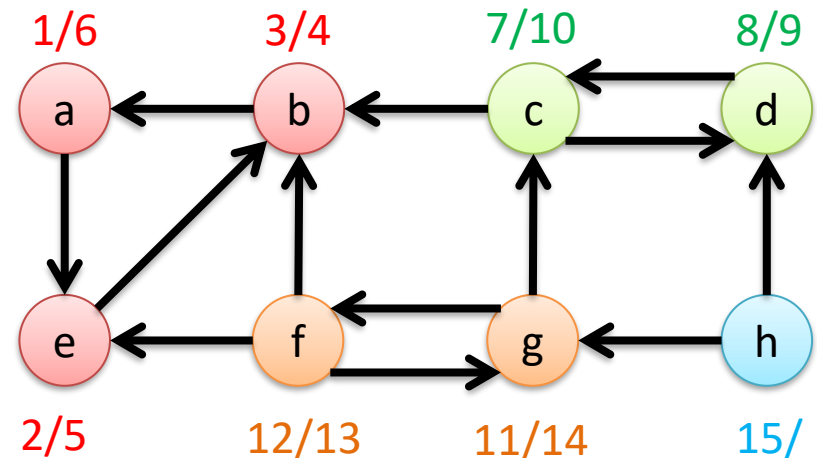
– Exemplo:



	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6

Arrows pointing up to the table:

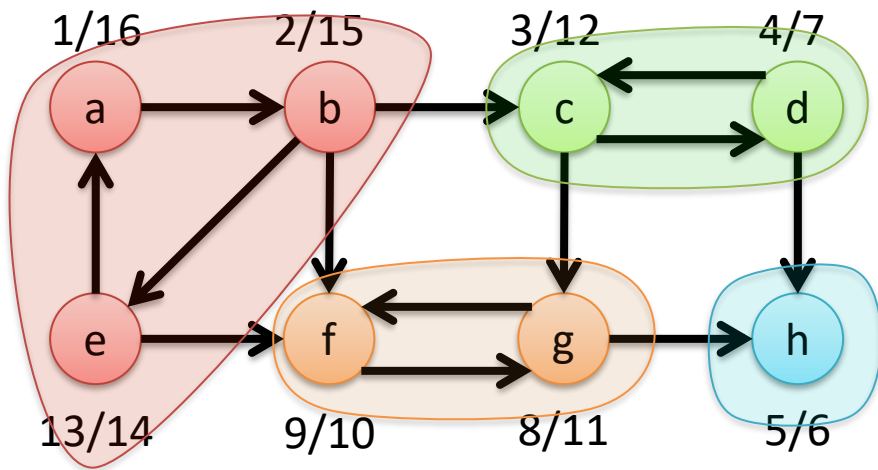
- Red arrow under 'a' (16)
- Green arrow under 'c' (12)
- Orange arrow under 'g' (11)
- Blue arrow under 'h' (6)



Algoritmos em grafos

- Componentes fortemente conectados

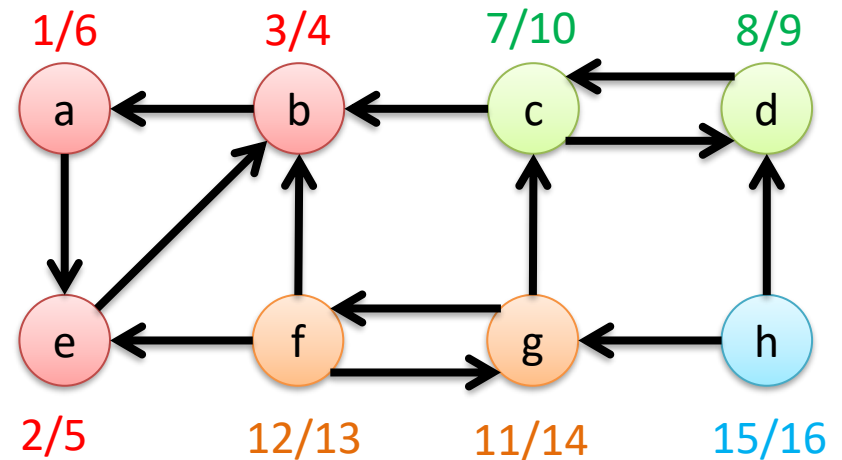
– Exemplo:



	a	b	c	d	e	f	g	h
f	16	15	12	7	14	10	11	6

Arrows pointing up to the table:

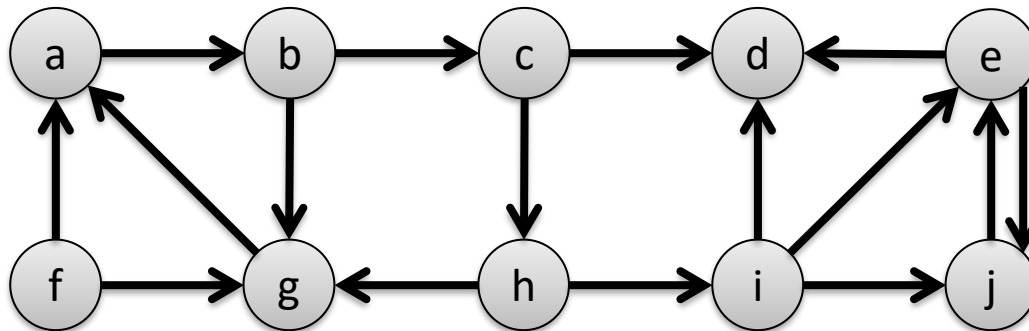
- Red arrow under 'a' (16)
- Green arrow under 'c' (12)
- Orange arrow under 'g' (11)
- Blue arrow under 'h' (6)



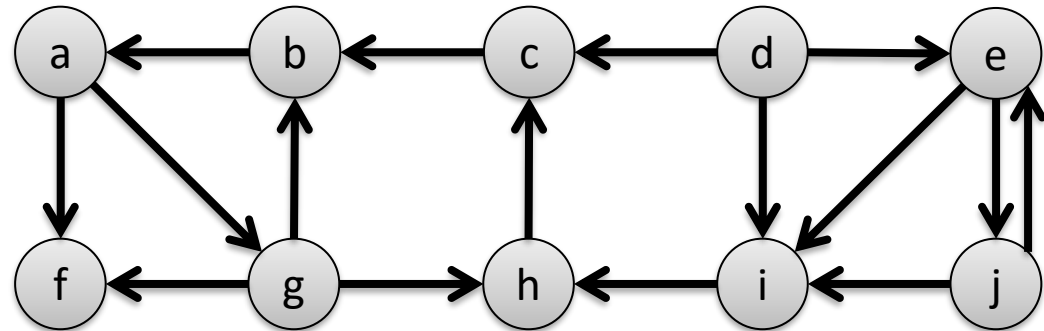
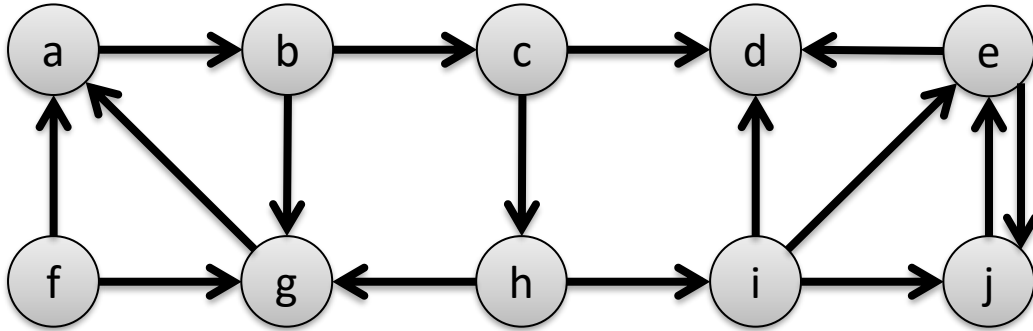
Algoritmos em grafos

- **Exercício 2**

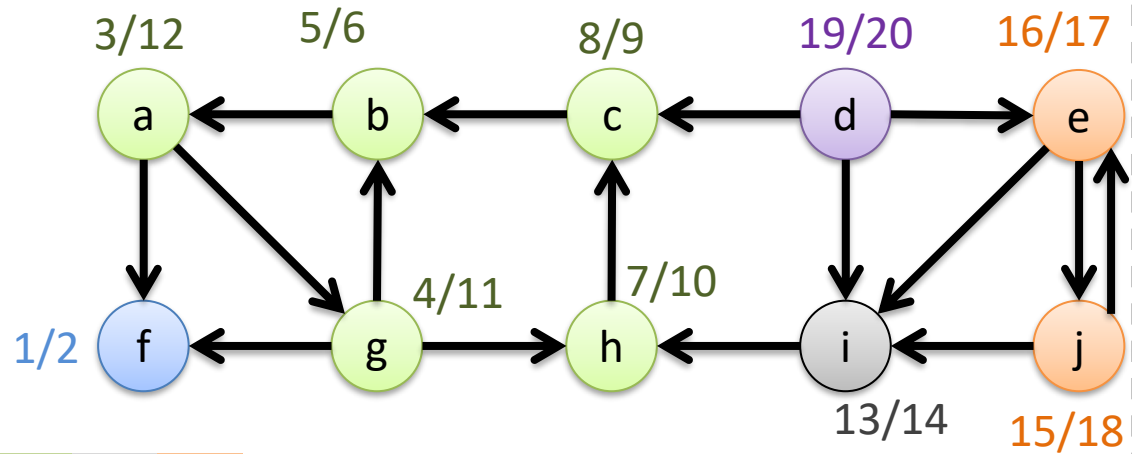
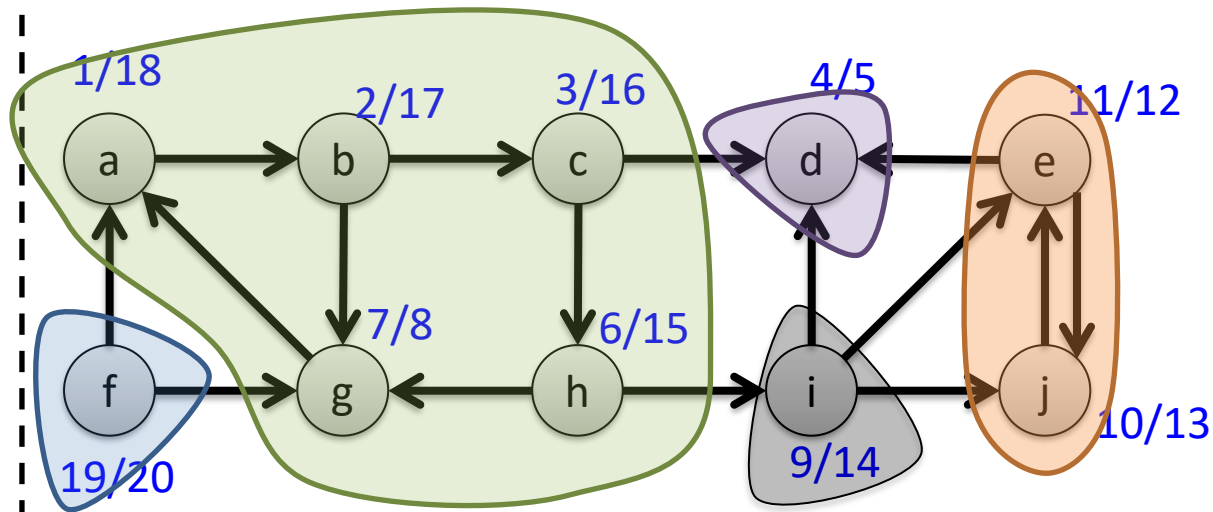
- Identifique os componentes fortemente conectados do grafo abaixo:



- **Exercício 2**

[illegible]

Exercício 2



	a	b	c	d	e	f	g	h	i	j
f	18	17	16	5	12	20	8	15	14	13

Algoritmos em grafos

FIM