TEG

Gilmário B. Santos

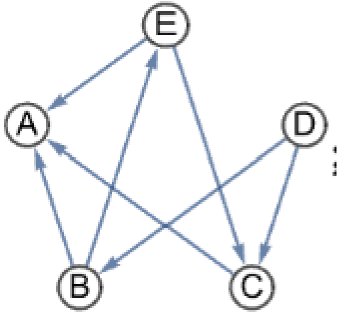
gilmario.santos@udesc.br

http://www.joinville.udesc.br/portal/pagina/gilmario

Ordenação topológica

A classificação topológica é uma maneira de organizar uma coleção de tarefas ou eventos em uma sequência, tal que cada tarefa venha antes das tarefas que dependem dela.

A ordenação determina a ordem de execução de um conjunto de tarefas relacionadas, garantindo que você não inicie uma tarefa até que todos os seus pré-requisitos ou dependências sejam concluídos.



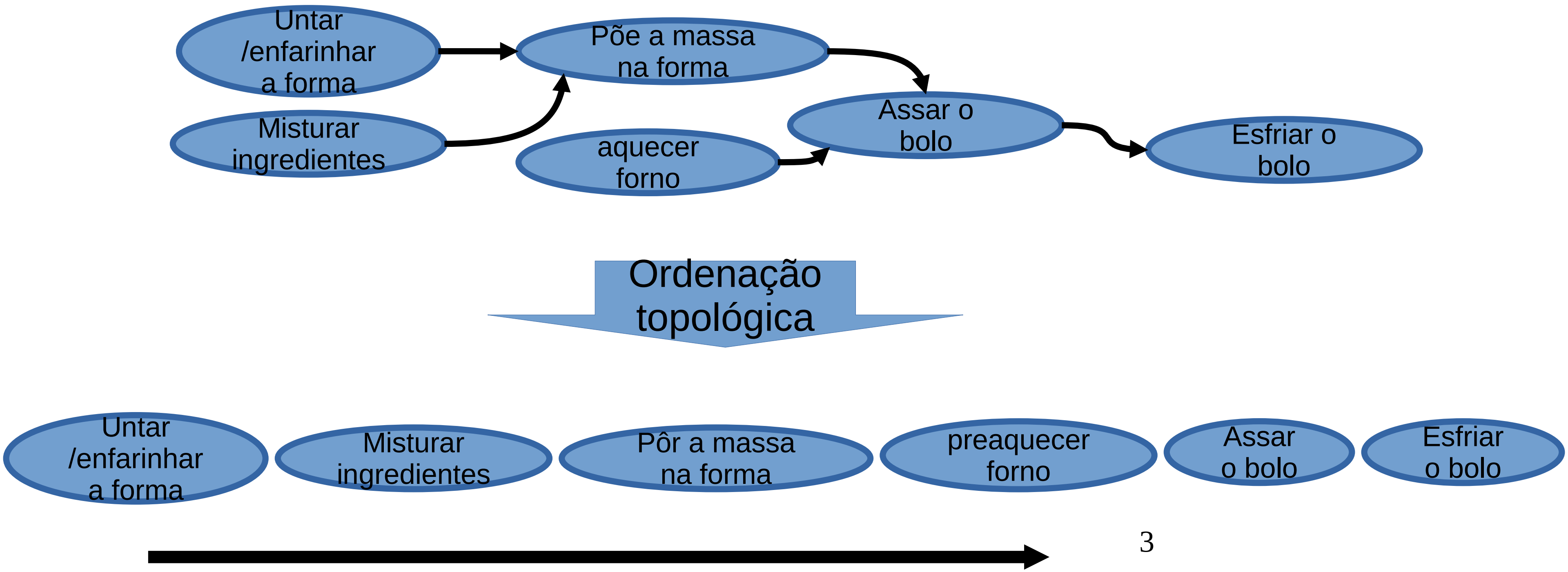
ordenação

{D, B, E, C, A}

Ordenação topológica

Por exemplo, para fazer bolo de chocolate:

1. Os ingredientes precisam ser misturados antes de irem para a forma;
2. A forma precisa ser untada e enfarinhada antes que a massa possa ser despejada;
3. O forno precisa ser preaquecido antes que o bolo possa assar;
4. O bolo precisa ser assado antes de esfriar;
5. O bolo precisa esfriar antes de poder ser coberto com glacê.



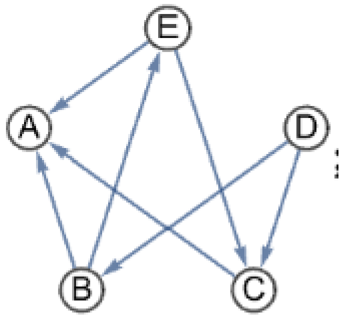
Ordenação topológica

Formalmente:

- Uma ordenação topológica é uma permutação *p* dos vértices de um grafo tal que uma aresta $\{v_i, v_j\}$ implica que v_i aparece antes de v_j em *p*.

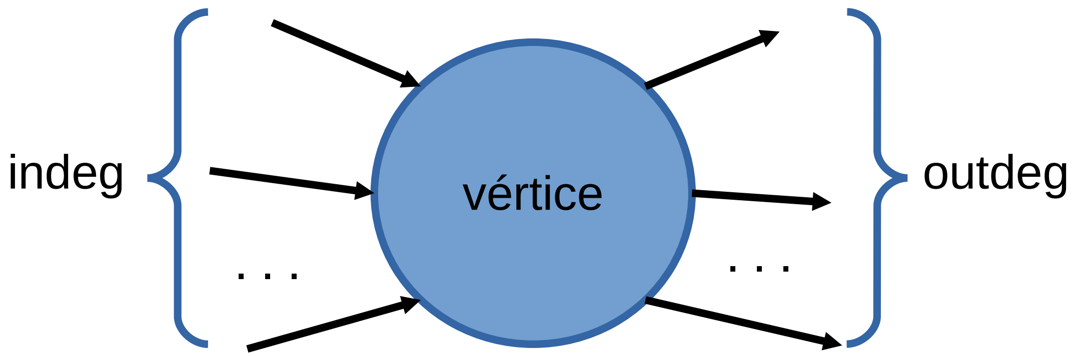
Apenas grafos acíclicos direcionados (ou *directed acyclic graphs - DAGs*) podem ser classificados topologicamente.

Um gráfico direcionado é um DAG se e somente se puder ser ordenado topologicamente, organizando os vértices como uma ordenação linear que seja consistente com todas as direções das arestas.



Ordenação topológica

Os vértices de um DAG apresentam grau de entrada (indeg) e grau de saída (outdeg)

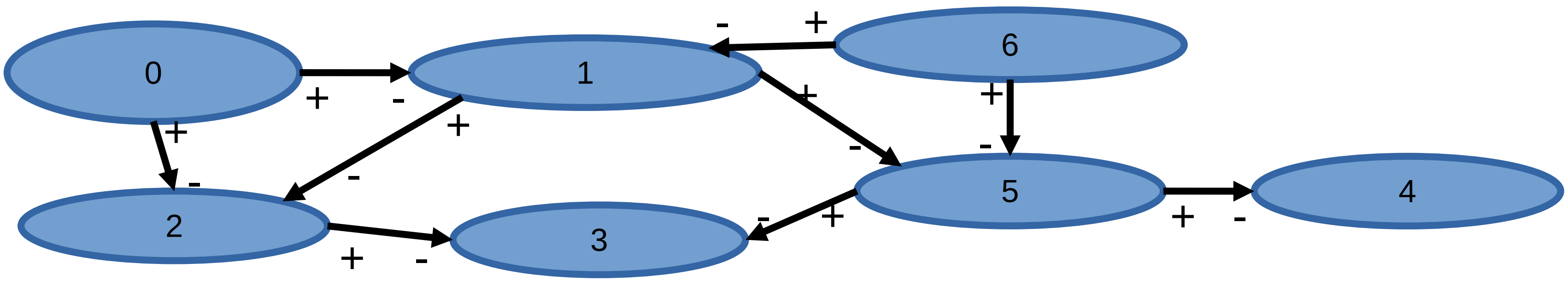


indeg pode ser igual a zero, como é o caso de uma tarefa que não depende de nenhuma outra.

Também pode ocorrer de *outdeg* ser igual a zero, é o caso para tarefas finais as quais não geram dependências.

Ordenação topológica

DAG:



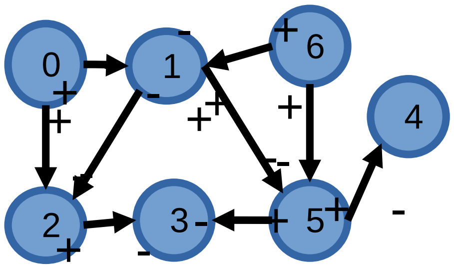
Matriz de Adjacências:

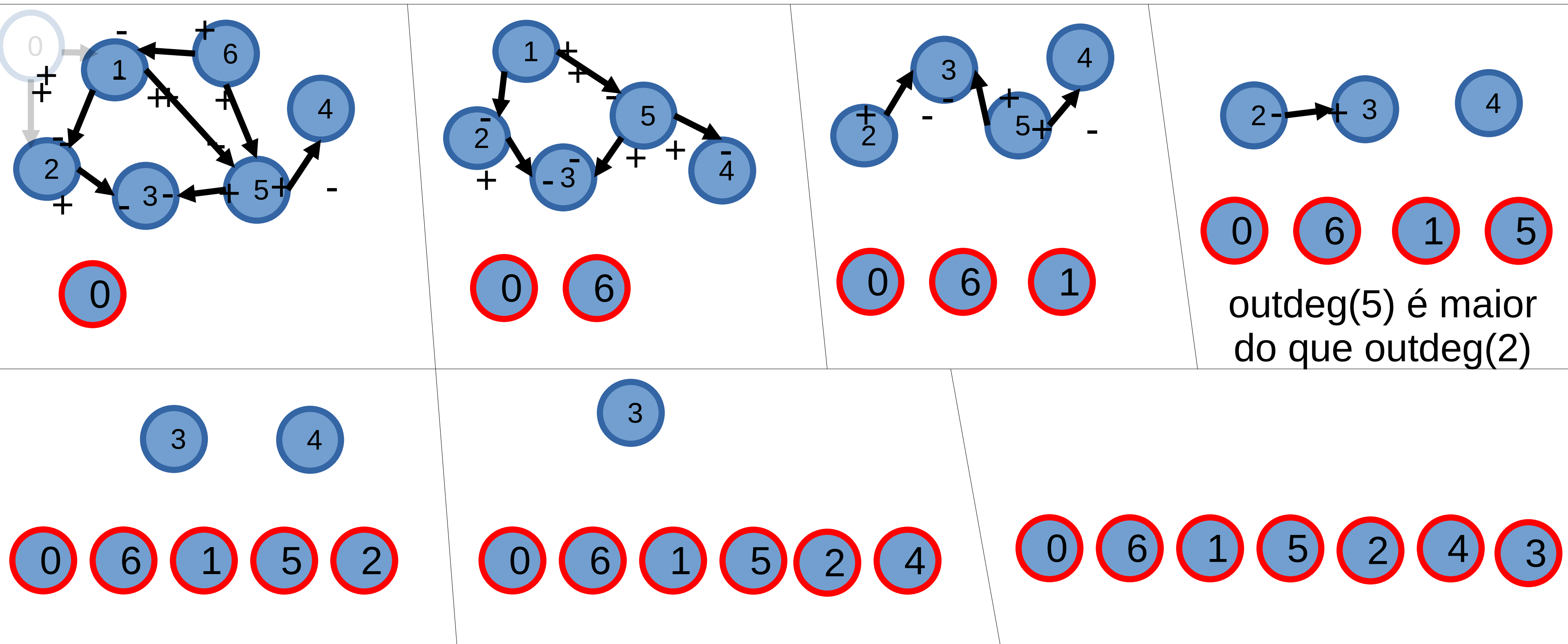
	0	1	2	3	4	5	6
0		+	+				
1	-		+			+	-
2	-	-		+			
3			-			-	
4						-	
5		-		+	+		-
6		+				+	

Ordenação topológica

Como extrair a ordenação?
Uma ideia simples para G(V,E) consiste em:
Para cada i em {1,2,3,4...|V|}

1. Identifique um nó sem arestas de entrada.
2. Adicione esse nó à ordenação.
3. Remova-o do grafo.





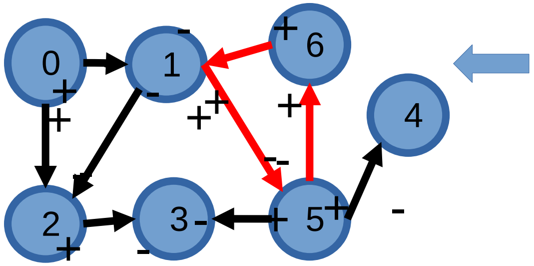
Ordenação topológica

Como extrair a ordenação?

Uma ideia simples para G(V,E):
Para cada i em {1,2,3,4...|V|}

1. Identifique um nó sem arestas de entrada.
2. Adicione esse nó à ordenação.
3. Remova-o do DAG.

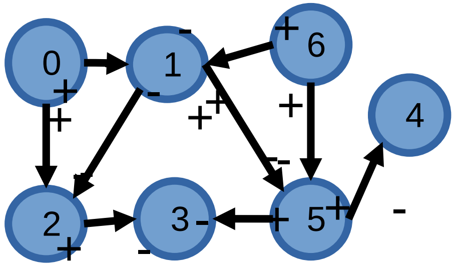
Se ao final do looping ainda restar um ou mais nós de indeg > 0, ou seja, se a ordenação topológica não contém todos os nós do grafo, então o grafo tem um ciclo e não existe nenhuma ordenação topológica para ele.



Se esse fosse o caso, o grafo não é seria DAG e não haveria ordenação

Ordenação topológica

Como extrair a ordenação?



Implementação mais realista:

Um pequeno ajuste para evitar remover os nós do grafo e destruir nossa entrada de dados:

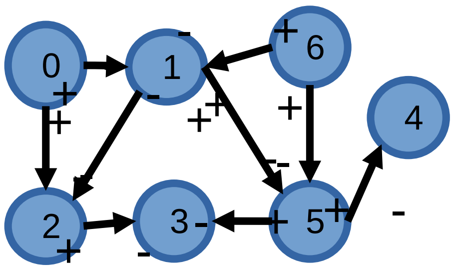
Precisaremos adaptar uma DFS usando uma fila para representar a ordenação topológica e uma tabela/mapa auxiliar para rastrear o grau de entrada e saída de cada nó.

Também precisaremos da matriz de adjacências.

Lembrando que, no grafo DAG, um nó *u* é vizinho de um nó *v* se existe aresta de saída de *v* para *u*.

Ordenação topológica

Como extrair a ordenação?



Construa

	0	1	2	3	4	5	6
0		+	+				
1	-		+			+	-
2	-	-		+			
3			-			-	
4						-	
5		-		+	+		-
6		+				+	

	Mapa auxiliar	
	indeg	outdeg
0	0	2
1	2	2
2	2	1
3	2	0
4	1	0
5	2	2
6	0	2

Ordenação topológica

Como extrair a ordenação?

Seja o grafo direcionado G(V,E):

Aplique

1. Existe um nó *v* nas condições abaixo?

    indeg(v)=0 **e**

    outdeg(v) máximo dentre os vértices cujo *indeg( )=0* **e**

    v não está na fila de ordenação

    SIM:

    2. Adicione *v* à fila da ordenação **e** decremente o *indeg* dos vizinhos de *v*.

    Existe um nó *u* vizinho de *v* nas condições abaixo?

    indeg(u)=0 **e**

    outdeg(u) é máximo nos vizinhos de *v;*

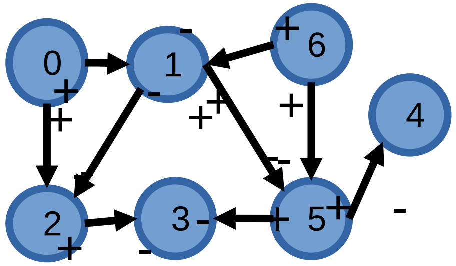
    SIM: faça *v=u* e vá ao passo 2

    NÃO: vá ao passo 1

    NÃO: então o grafo não é um DAG e não aceita ordenação

Ordenação topológica

Exercício com a implementação mais realista:



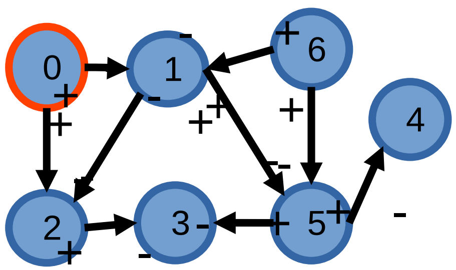
	Mapa auxiliar	
	indeg	outdeg
0	0	2
1	2	2
2	2	1
3	2	0
4	1	0
5	2	2
6	0	2

Fila vazia

0 ou 6 atendem ao passo 1 do algoritmo

Ordenação topológica

Exercício com a implementação mais realista:



	Mapa auxiliar	
	indeg	outdeg
0	0	2
1	2	2
2	2	1
3	2	0
4	1	0
5	2	2
6	0	2

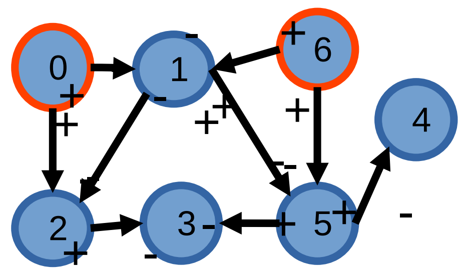
Fila
0

Insere na fila
decrementa indeg dos vizinhos

	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
1	1	2
2	1	1
3	2	0
4	1	0
5	2	2
6	0	2

Ordenação topológica

Exercício com a implementação mais realista:



	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
1	1	2
2	1	1
3	2	0
4	1	0
5	2	2
6	0	2

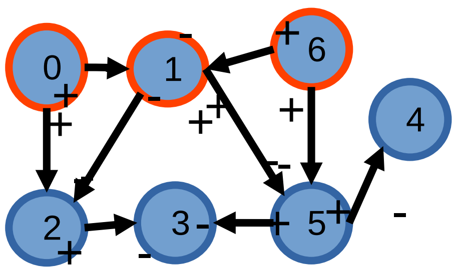
Fila
0 6

Insere na fila
decrementa indeg dos vizinhos

	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
1	0	2
2	1	1
3	2	0
4	1	0
5	1	2
~~6~~	~~0~~	~~2~~

Ordenação topológica

Exercício com a implementação mais realista:



	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
1	0	2
2	1	1
3	2	0
4	1	0
5	1	2
~~6~~	~~0~~	~~2~~

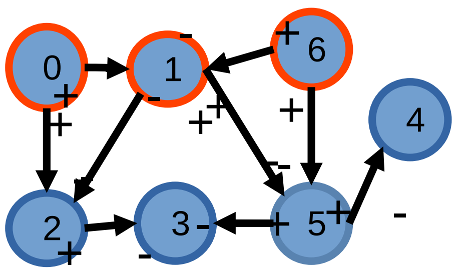
Fila
0 6 1

Insere na fila
decrementa indeg dos vizinhos

	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
2	0	1
3	2	0
4	1	0
5	0	2
~~6~~	~~0~~	~~2~~

Ordenação topológica

Exercício com a implementação mais realista:



	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
2	1	1
3	2	0
4	1	0
5	1	2
~~6~~	~~0~~	~~2~~

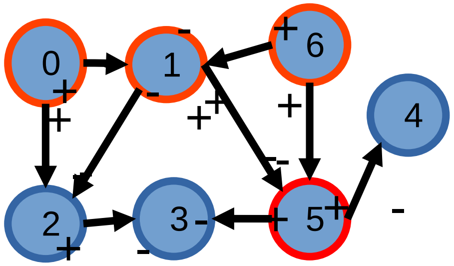
Fila

0 6 1

Vizinhos(1)={2, 5}, o nó 5 apresenta maior outdeg entre os vizinhos do nó 1

Ordenação topológica

Exercício com a implementação mais realista:



	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
2	1	1
3	2	0
4	1	0
5	1	2
~~6~~	~~0~~	~~2~~

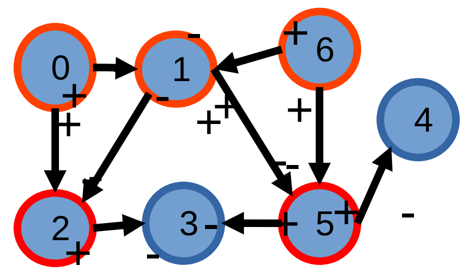
Fila
0 6 1 5

Insere na fila
decrementa indeg dos vizinhos

	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
2	0	1
3	1	0
4	0	0
~~5~~	~~0~~	~~2~~
~~6~~	~~0~~	~~2~~

Ordenação topológica

Exercício com a implementação mais realista:



	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
2	0	1
3	1	0
4	0	0
~~5~~	~~0~~	~~2~~
~~6~~	~~0~~	~~2~~

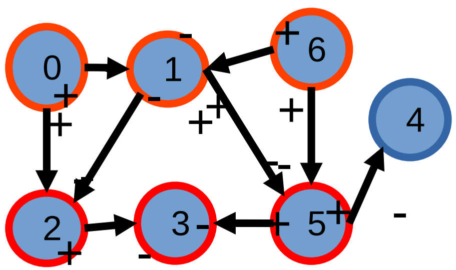
Fila
0 6 1 5 4

Insere na fila

	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
2	0	1
3	1	0
~~4~~	~~0~~	~~0~~
~~5~~	~~0~~	~~2~~
~~6~~	~~0~~	~~2~~

Ordenação topológica

Exercício com a implementação mais realista:



	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
2	0	1
3	1	0
~~4~~	~~0~~	~~0~~
~~5~~	~~0~~	~~2~~
~~6~~	~~0~~	~~2~~

Fila

0 6 1 5 4 2

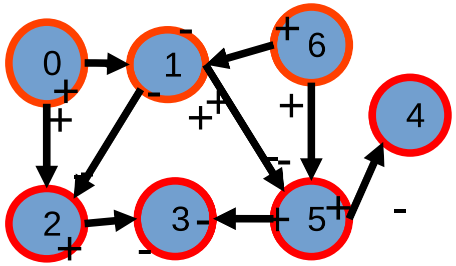
Insere na fila

decrementa indeg dos vizinhos

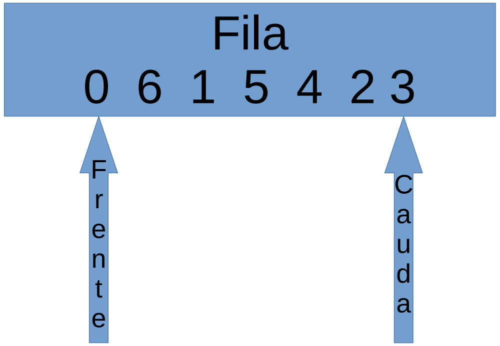
	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
~~2~~	~~0~~	~~1~~
3	0	0
~~4~~	~~0~~	~~0~~
~~5~~	~~0~~	~~2~~
~~6~~	~~0~~	~~2~~

Ordenação topológica

Exercício com a implementação mais realista:



	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
~~2~~	~~0~~	~~1~~
3	0	0
~~4~~	~~0~~	~~0~~
~~5~~	~~0~~	~~2~~
~~6~~	~~0~~	~~2~~



	Mapa auxiliar	
	indeg	outdeg
~~0~~	~~0~~	~~2~~
~~1~~	~~0~~	~~2~~
~~2~~	~~0~~	~~1~~
~~3~~	~~0~~	~~0~~
~~4~~	~~0~~	~~0~~
~~5~~	~~0~~	~~2~~
~~6~~	~~0~~	~~2~~

Ordenação topológica

Exercício:

1) Há um total de *7* cursos que você gostaria de realizar durante um congresso, os cursos são rotulados de *c0* a *c6* e existem pré-requisitos entre eles.

Suponha que você tenha recebido a lista de pré-requisitos onde *prerequisito[i] =* $[a_i, b_i]$ indicando que a_i é pré-requisito para b_i.

Por exemplo, o par [c0, c1], indica que para pegar o curso c1 você precisa passar no curso c0.

A lista de pré-requisitos é a seguinte:

[[c6,c1]; [c6,c5]; [c0,c1]; [c0,c2]; [c1,c2]; [c1,c5]; [c5,c3]; [c5,c4]; [c2,c3]]

Retorne uma ordenação válida de cursos que você deve fazer para finalizar todos os cursos.

Com os dados atuais, haveria alguma situação que impossibilitaria a realização de todos os cursos.

Ordenação topológica

Exercício:

2) Em relação ao exercício anterior, o certificado de participação é entregue para os que realizarem todos os 7 cursos. A organização conseguiu alocar três manhãs consecutivas, das 6:00 às 10:00, para a oferta dos 7 cursos, cada um deles com uma hora de duração.

Na tabela abaixo, as marcações (x) representam os pares de cursos que não podem ocorrer na mesma manhã, por questões de logística.

Diante desse “quadro” como devem ser distribuídos os cursos para que seja viável a certificação dos participantes obedecendo a ordem determinada na resposta à questão anterior? Justifique a sua resposta com base na coloração de vértices e na ordenação obtida.

	c0	c1	c2	c3	c4	c5	c6
c0			x	x	x		
c1			x	x			
c2	x	x		x		x	
c3	x	x	x			x	x
c4	x					x	x
c5			x	x	x		
c6				x	x		