

BCC204 - Teoria dos Grafos

Marco Antonio M. Carvalho

(baseado nas notas de aula do prof. Haroldo Gambini Santos)

Departamento de Computação
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto

11 de maio de 2016



Site da disciplina:

- ▶ <http://www.decom.ufop.br/marco/>

Moodle:

- ▶ www.decom.ufop.br/moodle

Lista de e-mails:

- ▶ bcc204@googlegroups.com

Para solicitar acesso:

- ▶ <http://groups.google.com/group/bcc204>

- 1 Ordenação Topológica
- 2 Busca em Profundidade - DFS
- 3 Algoritmo de Kahn
- 4 Aplicações

Definição

Uma **Ordenação Topológica** de um grafo acíclico direcionado (GAD ou DAG, da sigla em inglês), é uma ordenação linear de seus vértices, na qual cada vértice aparece antes de seus descendentes;

Em outras palavras, é uma ordenação linear de vértices na qual cada vértice **precede** o conjunto de vértices que formam seu fecho transitivo direto;

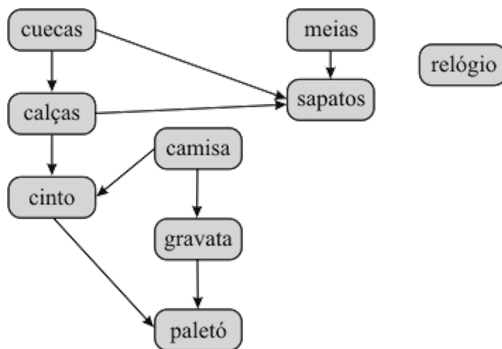
Cada GAD possui uma ou mais ordenações topológicas;

Caso um grafo possua ciclos, não é possível estabelecer uma relação de precedência entre os vértices, e portanto, é impossível estabelecer uma ordenação topológica.

Ordenação Topológica

Questionamento

É possível obter uma ordenação linear (por exemplo, em uma linha horizontal) das tarefas abaixo para de maneira a se vestir corretamente e respeitar todas as precedências entre as tarefas?



Histórico

Algoritmos de ordenação topológica começaram a ser estudados na década de 60, no contexto da técnica PERT (*Program Evaluation and Review Technique*) para alocação de tarefas em gerenciamento de projetos e geração de estimativas sobre a execução do mesmo:

- ▶ Tarefas a serem desempenhadas em um projeto são representadas por vértices;
- ▶ Existe um arco entre as tarefas v e w caso a tarefa v obrigatoriamente deve ser terminada antes que a tarefa w seja começada;
- ▶ Uma ordenação topológica dos vértices nos fornece uma maneira de realizar todas as tarefas sem violar as precedências entre tarefas.

Algoritmos

Existem diferentes algoritmos para obtenção de ordenações topológicas em grafos, os de melhor desempenho possuem complexidade linear;

Dois dos mais utilizados são o baseado na **Busca em Profundidade** (ou DFS) e o **Algoritmo de Kahn**.

Princípio

Atenção, este algoritmo considera o sentido oposto dos arcos em relação ao original apresentado nos diagramas!

O algoritmo considera que, caso um vértice x dependa de um vértice y , então existe um arco de x para y .

Terminologia

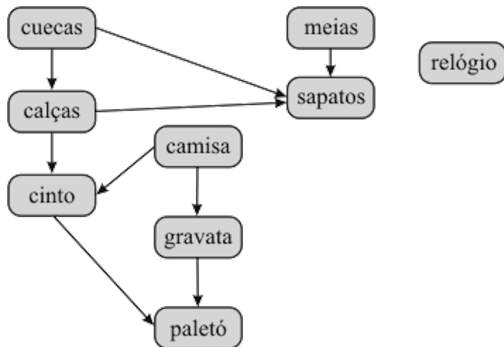
- ▶ L : Lista que conterá os elementos da ordenação topológica;
- ▶ Um vértice pode ser **não marcado**, **temporariamente marcado** ou **definitivamente marcado**;
- ▶ Inicialmente, todos os vértices são não marcados;
- ▶ Ao serem atingidos pela primeira vez, os vértices são temporariamente marcados;
- ▶ Após terem todas as suas dependências examinadas, os vértices são definitivamente marcados;
- ▶ Caso um vértice temporariamente marcado seja examinado novamente, o grafo possui pelo menos um ciclo.

Busca em Profundidade - DFS

Entrada: Grafo $G = (V, A)$

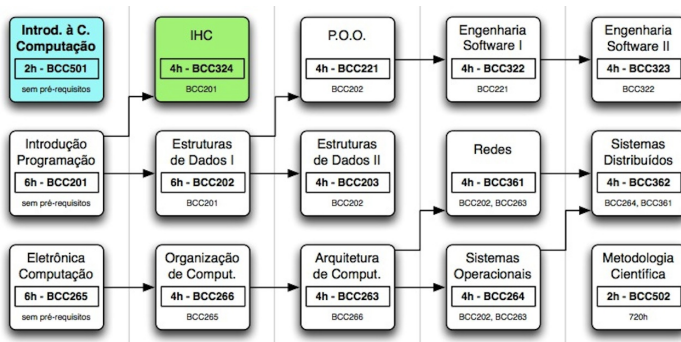
```
1  $L \leftarrow \emptyset$ ;  
2 enquanto existir vértice não marcado e sem arcos de entrada faça  
3   |   selecione um vértice  $v$  não marcado;  
4   |   visite( $G, v, L$ );  
5 fim  
  
1 função visite( $G, v, L$ )  
2 se  $v$  é temporariamente marcado então retorna Erro; // detecção de ciclo ;  
3 se  $v$  é não marcado então  
4   |   marque temporariamente  $v$ ;  
5   |   para cada arco ( $vw$ ) faça  
6   |   |   visite( $G, w, L$ ); // chamada recursiva da função  
7   |   fim  
8   |   marque definitivamente  $v$ ;  
9   |   adicione  $v$  ao final de  $L$ ;  
10 fim
```

Exemplo



Lembre-se de inverter o sentido dos arcos!

Exercício



Lembre-se de inverter o sentido dos arcos!

Aplicação

O algoritmo de *Kahn* data de 1962 e possui como princípio determinar a cada instante os vértices que não possuam arcos de entrada e inserir na solução;

A cada vértice inserido na solução, todos seus arcos correspondentes são removidas do grafo;

Detecta a existência de ciclos no grafo.

Terminologia

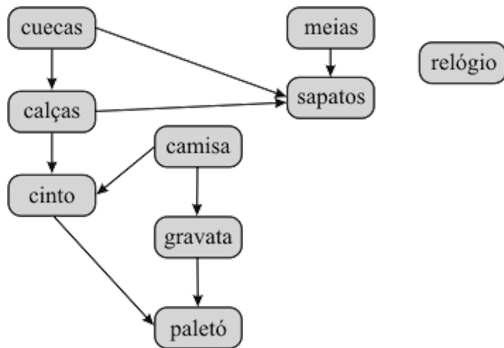
- ▶ L : Lista que conterá os elementos da ordenação topológica;
- ▶ S : Conjunto de vértices que não possuem arcos de entrada.

Algoritmo de *Kahn*

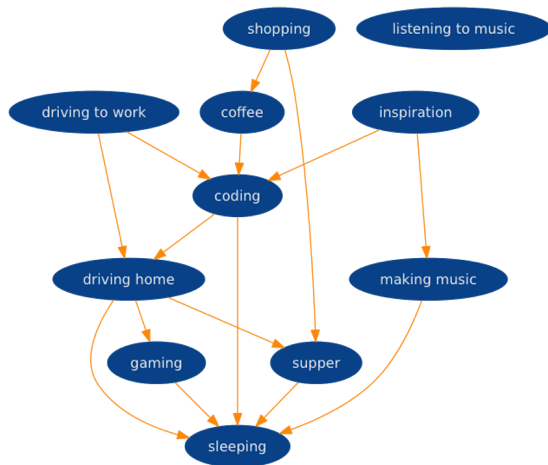
Entrada: Grafo $G=(V, A)$

```
1  $L \leftarrow \emptyset$ ;  
2  $S \leftarrow$  vértices sem arcos de entrada;  
3 enquanto  $S \neq \emptyset$  faça  
4   Remova um vértice  $v$  de  $S$ ;  
5   Insira o vértice  $v$  em  $L$ ;  
6   para cada arco  $(vw)$  faça  
7     Remova o arco  $vw$  de  $A$ ;  
8     se  $w$  não possui mais arcos de entrada então  
9       Insira o vértice  $w$  em  $S$ ;  
10    fim  
11  fim  
12 fim  
13 se  $A \neq \emptyset$  então retorna Erro; //o grafo possui pelo menos um ciclo ;  
14 senão retorna  $L$ ; //a ordenação topológica ;
```

Exemplo



Exercício 2



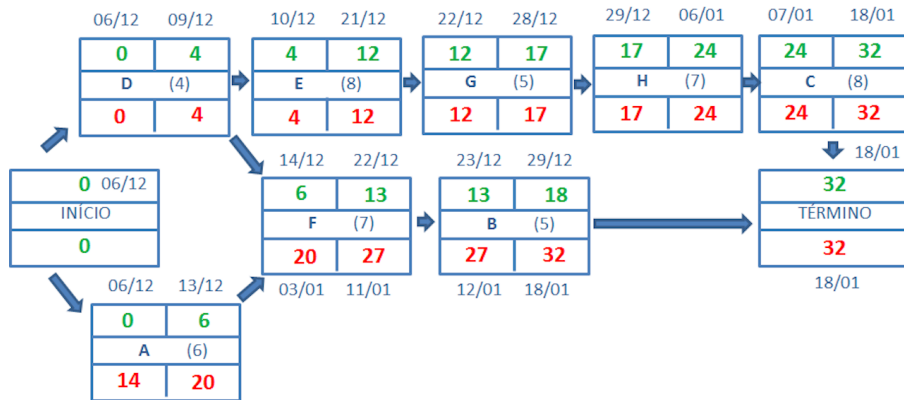
Exemplos

- ▶ Roteiros de instalação de pacotes;
- ▶ O *make* (do *makefile*);
- ▶ Confeção de dicionários;
- ▶ Organização de bancos de dados;
- ▶ Sistemas geográficos;
- ▶ Alocação de projetos (*Project Scheduling*).

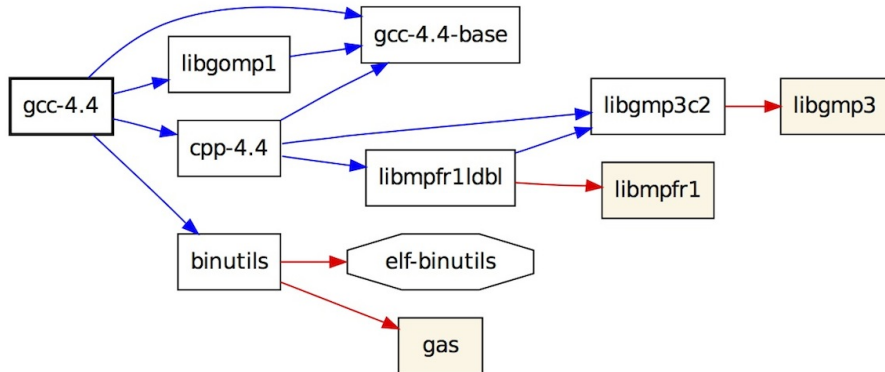
Aplicações: Caminho Crítico



Aplicações: Caminho Crítico (2)



Aplicações: Dependências do pacote GCC



Curiosidade

Em plataformas Linux, há o comando *tsort*, que realiza uma ordenação topológica em uma lista de adjacências informada pelo usuário.

Dúvidas?

