ESTRUTURAS DE DADOS II

MSC. DANIELE CARVALHO OLIVEIRA

DOUTORANDA EM CIÊNCIA DA COMPUTAÇÃO - USP

MESTRE EM CIÊNCIA DA COMPUTAÇÃO – UFU

BACHAREL EM CIÊNCIA DA COMPUTAÇÃO - UFJF

2

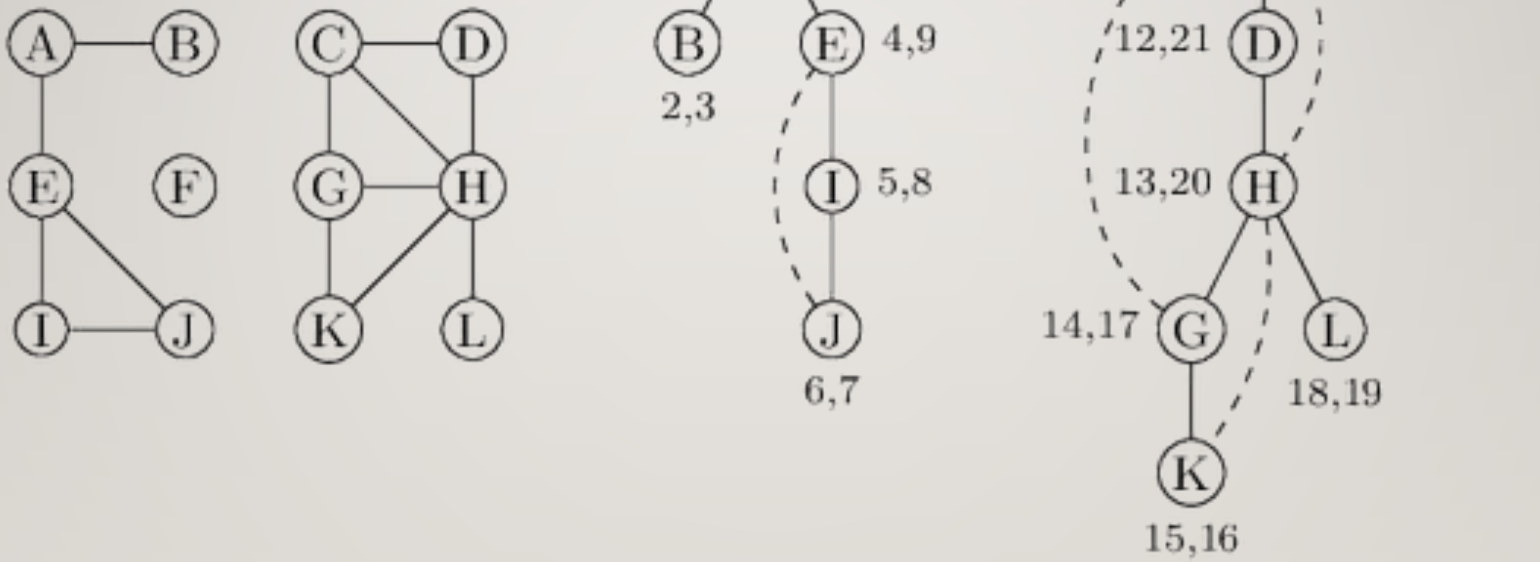
ORDENAÇÃO TOPOLÓGICA

3 ORDEM DE VISITAÇÃO

- Aplicando o algoritmo DFS é possível identificar o tempo de visitação de cada vértice de um grafo

- Deve-se identificar o momento em que um nó começa a ser explorado até o momento em que termina de ser explorado, utiliza-se para isso um contador
  - Denominamos esses pontos de pre-visited e pos-visited.

4

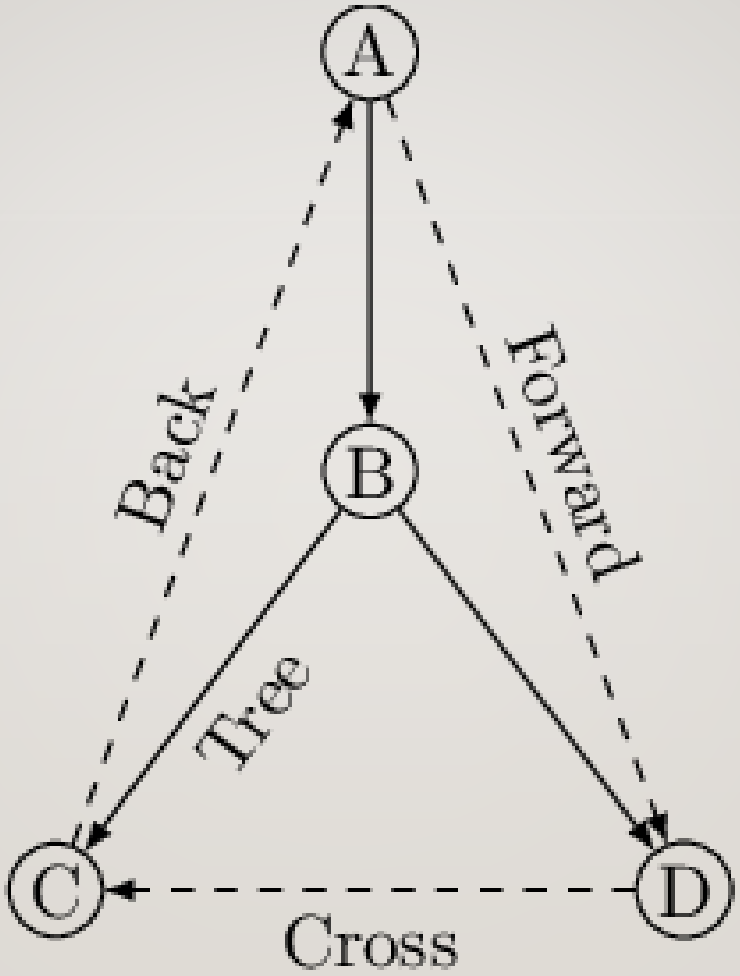

A
B
E
F
I
J
C
D
G
H
K
L
1,10
A
B
E
4,9
2,3
I
5,8
J
6,7
11,22
C
23,24
F
12,21
D
13,20
H
14,17
G
L
18,19
K
15,16

5 PROPRIEDADES

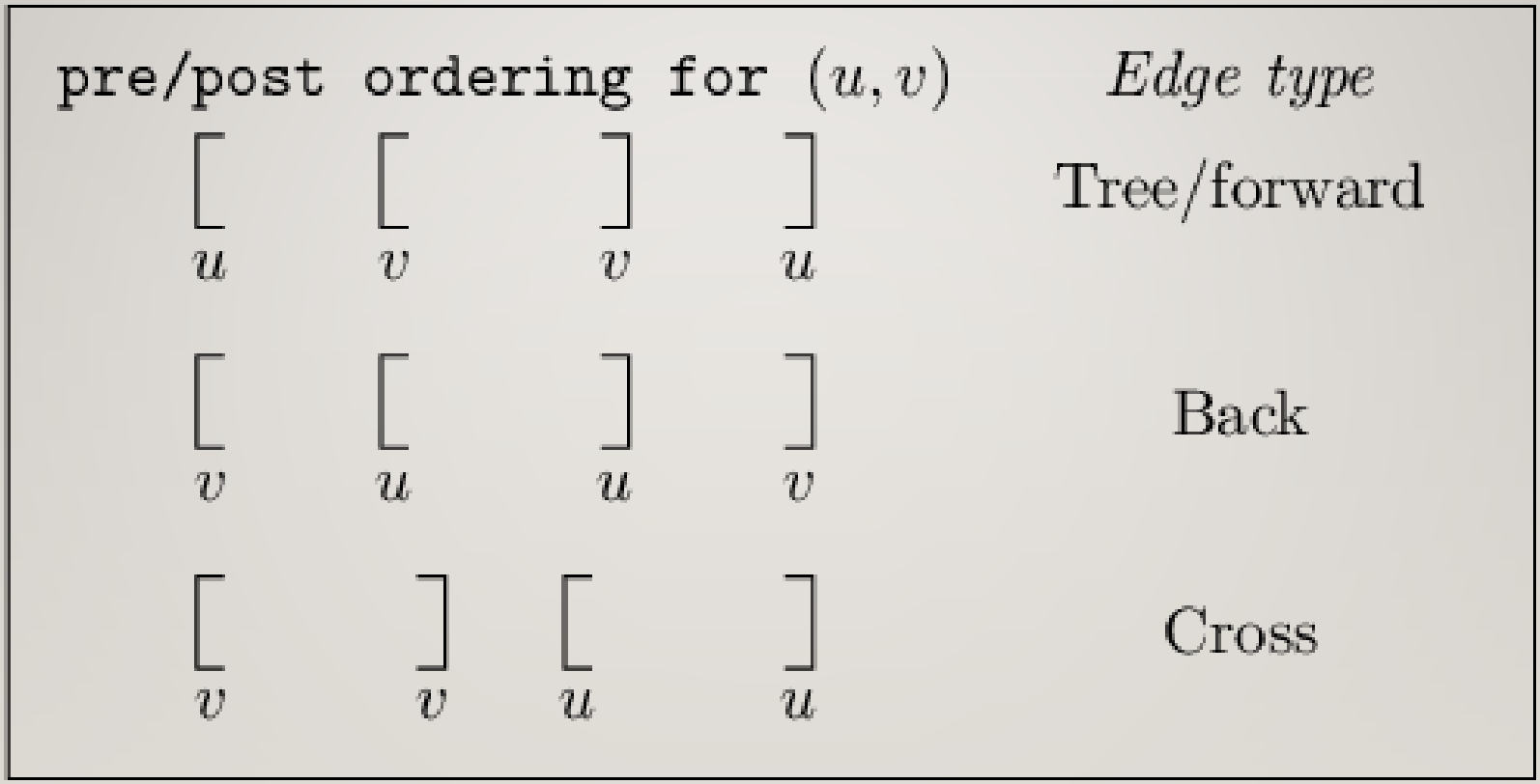
- Grafos Não Dirigidos
  - Arestas de árvore (tree-edges)
  - Arestas Não árvore (back-edges)

- Digrafos
  - Tree-edges – é parte do caminho do DFS
  - Forward-edges – levam a um nó descendente na árvore (não filho)
  - Backward-edges – leva a um antecedente já explorado na árvore
  - Cross-edges – leva a um nó que não é nem antecedente nem descendente, mas que já foi explorado

6



7


pre/post ordering for (u, v)
Edge type
u v v u
Tree/forward
v u u v
Back
v v u u
Cross

8 TEOREMA

Um grafo dirigido é acíclico, se e somente se, o grafo não possui back edges

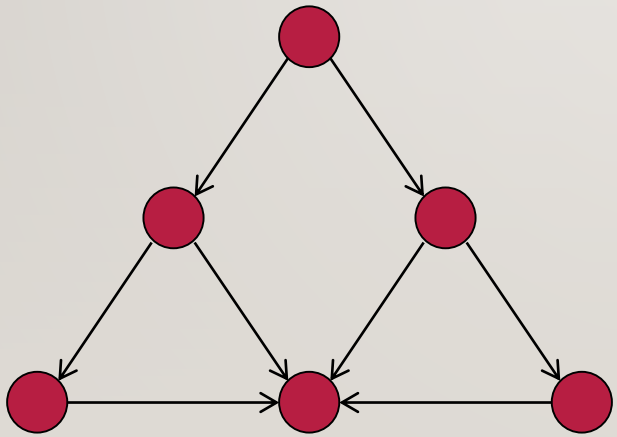
9 LINEARIZAÇÃO

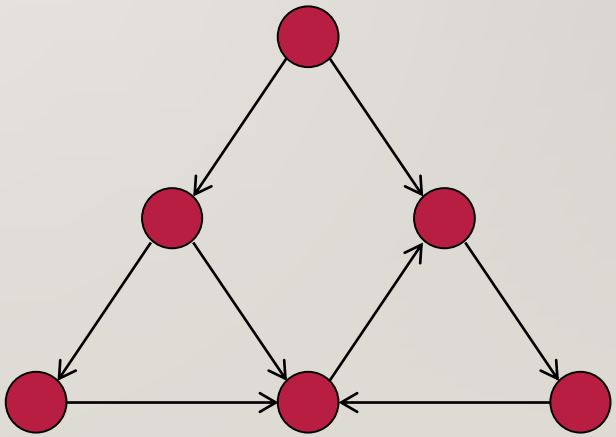
- Seja G um grafo qualquer, uma linearização de G é uma sequencia $< u_1, u_2, \dots, u_n >$, com $\{u_1, u_2, \dots, u_n\} \in V$ e $(u_i, u_{i+1}) \in E$

- Um grafo G é linearizável, se e somente se, G é acíclico

10 ORDENAÇÃO TOPOLÓGICA

- Consideremos grafos dirigidos acíclicos
- Um digrafo é acíclico se não tem ciclos. Digrafos acíclicos também são conhecidos como DAGs (= directed acyclic graphs)





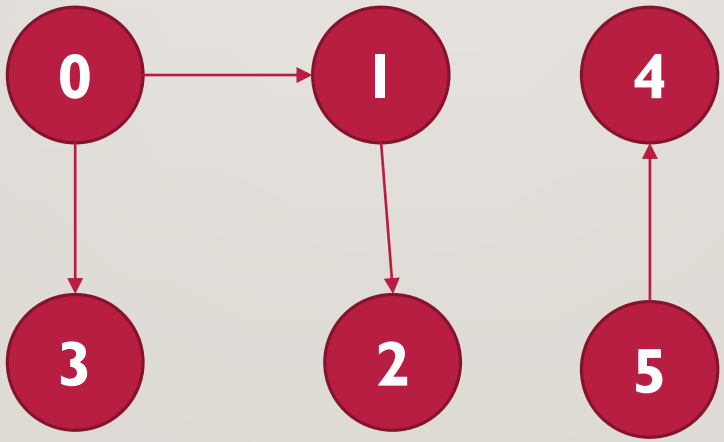
DIGRAFO ACÍCLICO

- Digrafos que não tem ciclos, como:
  - Hierarquia de herança entre classes em orientação a objetos
  - Pré-requisitos entre disciplinas
  - Restrições de cronograma entre tarefas de um projeto

- Toda árvore direcionada é um digrafo acíclico
- Todo caminho num digrafo acíclico é simples, não tem repetição de vértices
- Como saber se um digrafo é acíclico?

12 ORDENAÇÃO TOPOLÓGICA

- Grafos direcionados acíclicos podem ser usados para indicar precedência de eventos
  - O evento 1 só pode ocorrer após o evento 0
  - O evento 4 pode ocorrer antes do evento 0



13 ORDENAÇÃO TOPOLÓGICA

- É uma ordenação nos vértices de forma que todas as arestas vão da esquerda para a direita.

- Com a ordenação Topológica:
  - Verificar se um grafo é bicolorível
  - Detecção de ciclos
  - Caminhos Mínimos
  - Conectividade

14 ORDENAÇÃO TOPOLÓGICA

O grafo abaixo tem diversas ordenações topológicas possiveis:

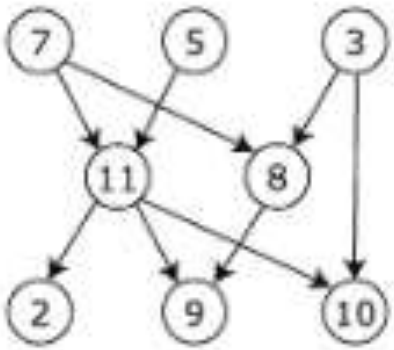
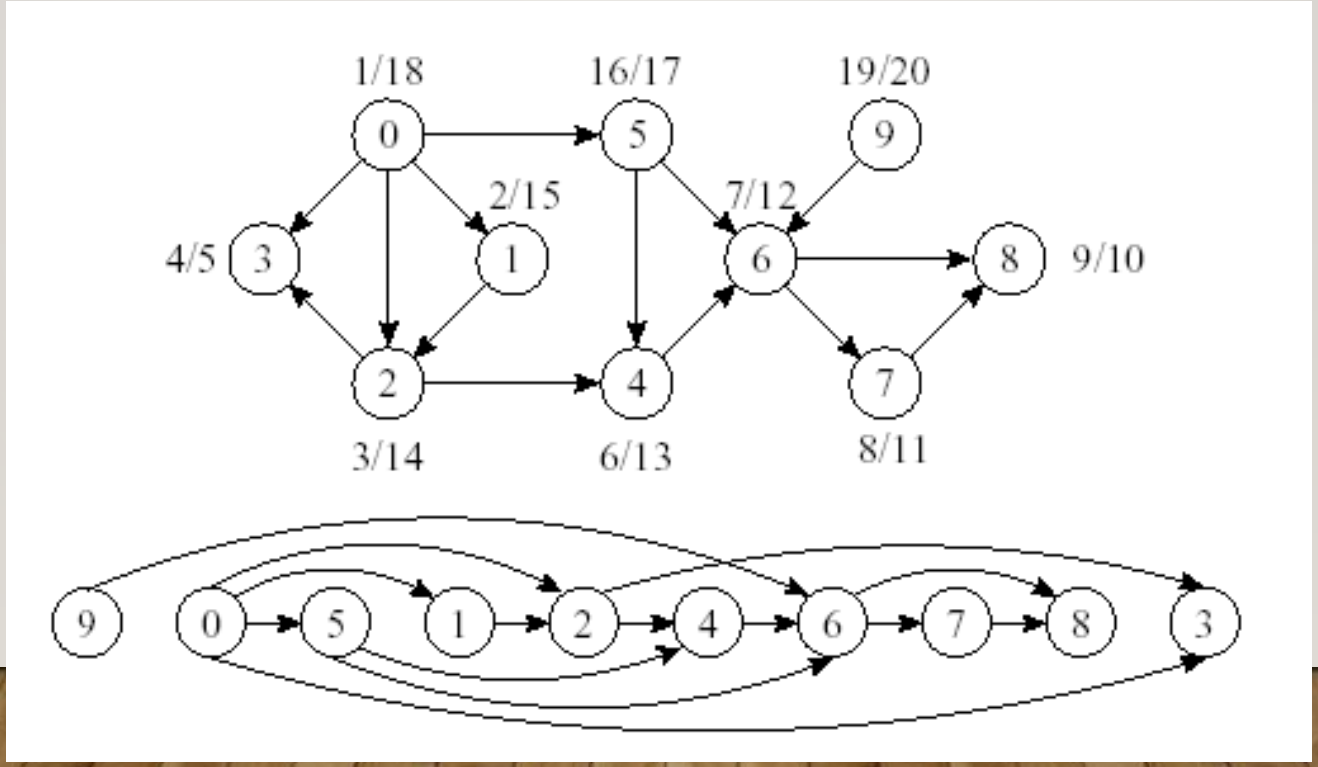


Figura por Derrick Coetzee

- 7, 5, 3, 11, 8, 2, 9, 10 (visual esquerda-para-direita, de-cima-para-baixo)
- 3, 5, 7, 8, 11, 2, 9, 10 (vértice de menor número disponível primeiro)
- 3, 7, 8, 5, 11, 10, 2, 9
- 5, 7, 3, 8, 11, 10, 9, 2 (menor número de arestas primeiro)
- 7, 5, 11, 3, 10, 8, 9, 2 (vértice de maior número disponível primeiro)
- 7, 5, 11, 2, 3, 8, 9, 10

15 ORDENAÇÃO TOPOLÓGICA

- Uma aresta direcionada (u, v) indica que a atividade u tem que ocorrer antes da atividade v.



16 ALGORITMOS DE ORDENAÇÃO TOPOLÓGICA

- Ao receber um Digrafo G, um algoritmo de ordenação topológica deve devolver:
  - Uma ordenação topológica de G, ou
  - Um ciclo de G.

ALGORITMO DE ORDENAÇÃO TOPOLÓGICA : ELIMINAÇÃO DE FONTES

- Algoritmo de Kahn
  - Encontra os vértices "fonte" (com grau de entrada zero) e os insere em um conjunto S (uma fila)
    - Ao menos um vértice desses deve existir se o grafo é acíclico
  - Partindo do princípio que, se os vértices fonte e seus arcos de saída forem removidos, o grafo remanescente é dígrafo acíclico
    - Remove da fila sucessivamente os vértices fontes
    - Rotula-os em ordem de remoção e remove seus arcos

18 ALGORITMO DE KAHN

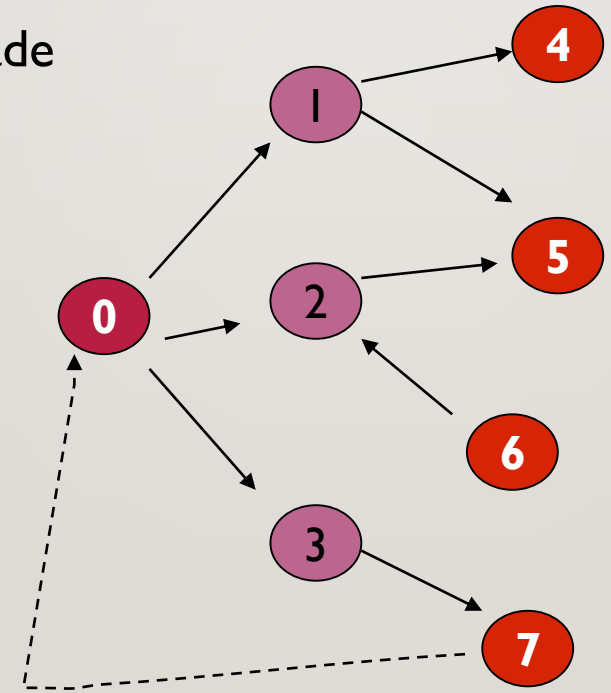
```
L ← Lista vazia que irá conter os elementos ordenados
S ← Conjunto de todos os nós fonte

enquanto S é não-vazio faça
        remova um nó n de S
        insira n em L
        para cada nó m com uma aresta de n até m faça
                remova a aresta do grafo
                se m não tem mais arestas de entrada então
                        insira m em S
se o grafo tem arestas então
  escrever mensagem de erro (grafo tem pelo menos um ciclo)
senão
  escrever mensagem (ordenação topológica proposta: L)
```

19 ALGORITMO DE TARJAN

- Se uma aresta de retorno é encontrada na DFS então o grafo possui um ciclo.
- Um grafo é acíclico sse na DFS não for encontrada nenhuma aresta de retorno.
- Busca em profundidade é utilizada para determinar se o grafo orientado é acíclico e então determinar uma ordenação topológica

ORDENAÇÃO TOPOLÓGICA

- Um grafo orientado é acíclico se, e somente se, não são encontradas arestas de retorno durante uma busca em profundidade



21 ALGORITMO DE TARJAN

- Iniciando a visita em v, visite todos os seus adjacentes (v, w) chamando a função DFS recursivamente para w.

- Após finalizar a lista de adjacências de cada vértice v sendo processado, adicione-o na ordem topológica.

22 ORDENAÇÃO TOPOLÓGICA

- Algoritmo
  1. Calcular o tempo de término de cada vértice utilizando a busca em profundidade
  2. A medida que é calculado seu tempo de término, insere-se o vértice no inicio de uma lista
- Os vértices ordenados topologicamente aparecem em ordem inversa aos seus tempos de término

23 ALGORITMO DE TARJAN

```
L ← Lista vazia que irá conter os elementos ordenados
S ← Conjunto de todos os nós sem arestas de entrada

função visita(nodo n)
    se n não foi visitado ainda então
        marque n como visitado
        para cada nodo m com uma aresta de n para m faça
            visite(m)
        adicione n em L

para cada nodo n em S faça
    visite(n)
```

Extra

- URI:
  - 1128; 1442; 1677; 1792; 1903

FIM DA AULA 16

Próxima aula:

Grafos: Árvore Geradora Mínima