

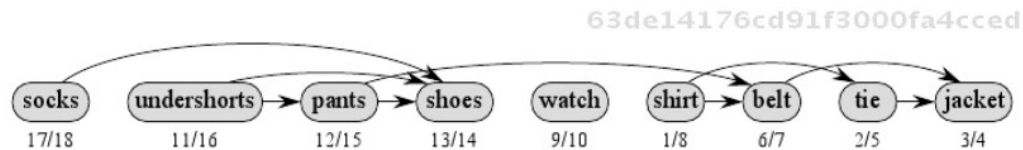
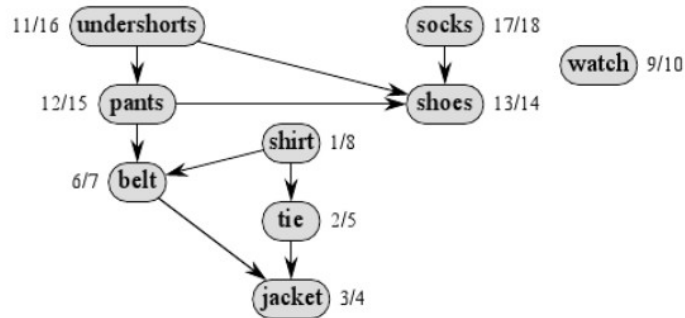
1001109 – Estruturas de Dados Avançadas

## Aula 10

# Ordenação Topológica

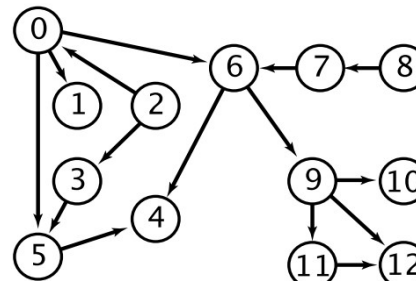
- Ordenação topológica é uma ordenação linear de todos os vértices, tal que se  $G$  contém um arco  $(u, v)$  então  $u$  aparece antes de  $v$ 
  - Logo, estamos falando de um DAG (grafo direcionado sem ciclos)
- Pode ser “vista” como uma ordenação de seus vértices ao longo de uma linha horizontal de tal forma que todas as arestas estão direcionadas da esquerda para a direita.

# Ordenação Topológica

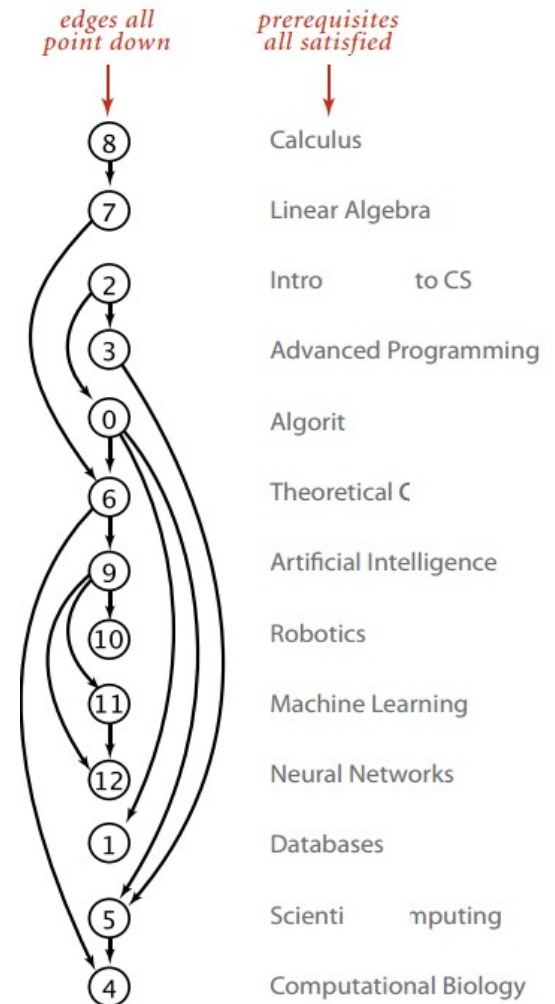


| application            | vertex     | edge                  |
|------------------------|------------|-----------------------|
| <i>job schedule</i>    | job        | precedence constraint |
| <i>course schedule</i> | course     | prerequisite          |
| <i>inheritance</i>     | Java class | extends               |
| <i>spreadsheet</i>     | cell       | formula               |
| <i>symbolic links</i>  | file name  | link                  |

Typical topological-sort applications



Standard digraph model



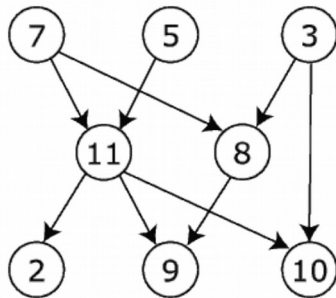
Topological sort

# Ordenação Topológica

- A busca em profundidade (DFS) pode ser utilizada para obter uma ordenação topológica, com uma pequena adaptação:
  - 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 ordenação topológica (cabeça de uma lista)

# Ordenação Topológica

- A complexidade do algoritmo de ordenação topológica em um DAG é a mesma da busca em profundidade, ou seja:
  - $O(|V|+|V|^2)$  para matrizes de adjacência, e;
  - $O(|V|+|A|)$  para listas de adjacência.
- Inserir um elemento na cabeça da lista é  $O(1)$ .
- A ordenação topológica não necessariamente é única:



- 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

# Ordenação Topológica

## Algoritmo de Kahn

```
int TopologicalSort(G, tsL) {  
    S = newQueue(); // fila S com vertice fonte  
    for (all vertices v in G) {  
        if (deg(v) == 0)  
            enqueue(S,v); // insere na fila S  
    }  
    t = 0; // inicia contador  
    while (!isEmpty(S)) {  
        v = dequeue(S); // retira fonte de S  
        v.topsort = t; // marca ordenacao  
        tsL[t++] = v; // insere em arranjo ordenado  
        for (all incident vertices (v,w)) {  
            w.inDegree--; // reduz grau de entrada dos adjacentes  
            if (w.inDegree == 0)  
                enqueue(S,w) // quando forem fontes, inserir em S  
        }  
    }  
}
```

