por Isadora de Oliveira

* Ordenação Topológica

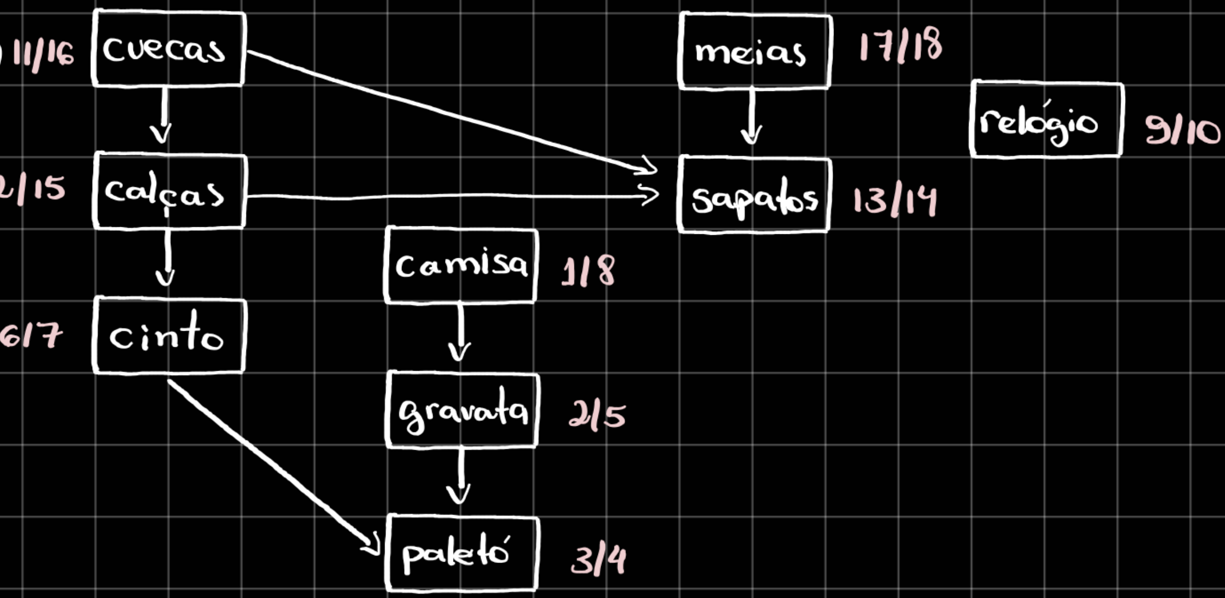
Usamos busca em profundidade para executar uma ordenação topológica de um grafo acíclico dirigido ou "gad". Uma ordenação topológica de um gad G=(V,E) é uma ordenação linear de todos os seus vértices, tal que se G contém uma aresta (u, v), então u aparece antes de v na ordenação. (Se o grafo contém um ciclo, nenhuma ordenação topologica é possível).
Ordenação topológica de um grafo como uma ordenação de seus vertices ao longo de uma linha horizontal de modo tal que todas as arestas dirigidas vão da esquerda para a direita.

Abaixo temos um exemplo que surge quando um professor se veste pela manhã. O professor deve vestir certas peças de roupa antes de outras (por exemplo, meias antes de sapato).

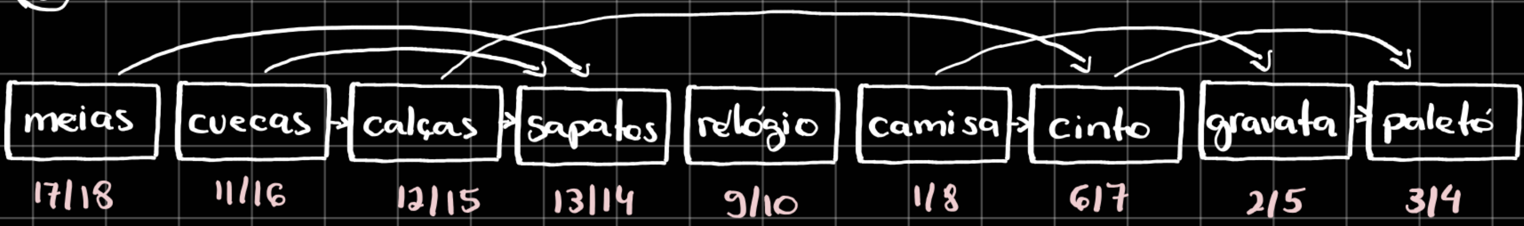
Uma aresta dirigida (u, v) no gad indica que a peça de roupa u deve ser vestida antes da peça v. Portanto, uma ordenação topológica desse gad dá uma ordem para o processo de se vestir. No exemplo abaixo temos o gad topologicamente ordenado como uma ordenação de vértices ao longo de uma linha horizontal tal que todas as arestas dirigidas vão da esquerda para a direita.

Exemplo de Ordenação topológica

(a)



(b)



Observamos que os vértices topologicamente ordenados aparecem na ordem inversa de seu tempo de término.

Ordenação topológica no tempo $\Theta(V+E)$ já que a busca em profundidade demora o tempo $\Theta(V+E)$ e que inserir cada um dos $|V|$ vértices 'a frente da lista ligada leva o tempo $O(1)$.

O seguinte algoritmo ordena topologicamente um gad:

TOPOLOGICAL-SORT

1 chama DFS(G) para calcular o tempo de termino $v.f$ para cada vértice v

2 à medida que cada vértice é terminado, inserir o vértice à frente de uma lista ligada

3 return a lista ligada de vértices

Lema 22.11

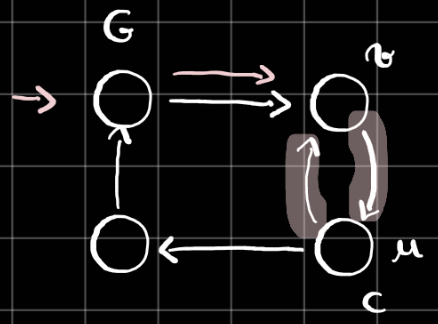
Duvidas:
As duas provas, não estão provando ser contratidório?

Um grafo dirigido G é acíclico se somente se uma busca em profundidade G não produz nenhuma aresta de retorno.

Prova

$\Rightarrow$ Suponha que uma busca em profundidade produza uma aresta de retorno (u, v). Então o vértice v é um ancestral do vértice u na floresta em profundidade. Assim, G contém um caminho de v a u, e a aresta de retorno (u, v) completa um ciclo.

$\Leftarrow$ Suponha que G contenha um ciclo c. Mostramos que uma busca em profundidade de G produz uma aresta de retorno. Seja v o primeiro vértice a ser descoberto em c e seja (u, v) a aresta precendente em c. No tempo $v.d$ os vértices de c formam um caminho de vértice brancos de v a u. Pelo teorema do caminho branco, o vértice u se torna um descendente de v na floresta em profundidade. Então (u, v) é uma aresta de retorno.



Se v o primeiro a ser descoberto temos (v, u) u se torna descendente de v logo se temos (u, v) isso seria uma aresta de retorno.

Teorema 22.12

TOPOLOGICAL-SORT produz uma ordenação topológica de um grafo acíclico dirigido dado como sua entrada.

Prova Suponha que DFS seja executado em determinado gad $G=(V,E)$ para determinar o tempo de término para seus vértices. É suficiente mostrar que para qualquer par de vértice distintos $u, v \in V$, se G contém uma aresta de u a v, então $v.f < u.f$. Considere qualquer aresta (u,v) explorada por DFS(G). Quando essa aresta é explorada v não pode ser cinzento já que nesse caso v seria um ancestral de u e (u,v) seria uma aresta de retorno o que contradiz o Lema 22.11. Portanto v deve ser branco ou preto. Se v é branco ele se torna um descendente de u e, assim $v.f < u.f$. Se v é preto, ele já terminou, de modo que $v.f$ já foi definido. Como ainda estamos explorando u, ainda temos de atribuir um carimbo de tempo a $u.f$ e, tão logo o façamos, também teremos $v.f < u.f$ Assim para qualquer aresta (u,v) no gad, temos $v.f < u.f$, o que prova o teorema.