

Ordenação topológica

Algoritmos em Grafos

Marco A L Barbosa



Este trabalho está licenciado com uma Licença Creative Commons - Atribuição-Compartilhual 4.0 Internacional.

Conteúdo

Introdução

Procedimento topological-sort

Exemplo de execução

Análise do tempo de execução do topological-sort

Corretude do topological-sort

Referências

O estudo utilizando apenas este material **não é suficiente** para o entendimento do conteúdo. Recomendamos a leitura das referências no final deste material e a resolução (por parte do aluno) de todos os exercícios indicados.

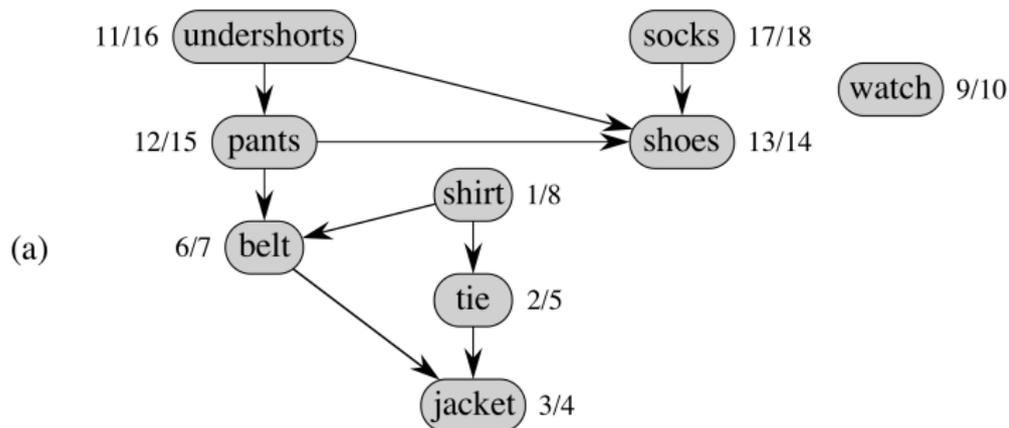
Introdução

Introdução

- ▶ Uma **ordenação topológica** de um grafo acíclico orientado $G = (V, E)$ é uma ordenação linear de todos os vértices, tal que para toda aresta $(u, v) \in E$, u aparece antes de v na ordenação
- ▶ Se os vértices forem dispostos em uma linha horizontal, todas as arestas devem ter a orientação da esquerda para direita
- ▶ Aplicação
 - ▶ Definição da ordem de execução de tarefas dependentes. Ex: `Makefile`

Introdução

- ▶ Exemplo: o professor Bumstead deve se vestir pela manhã



Procedimento topological-sort

Procedimento topological-sort

topological-sort(G)

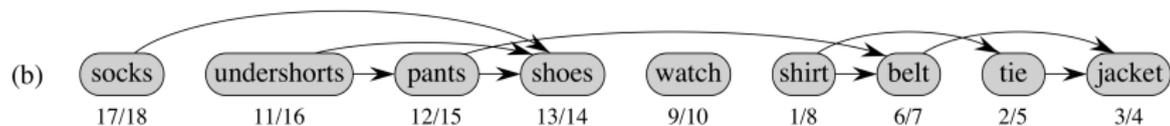
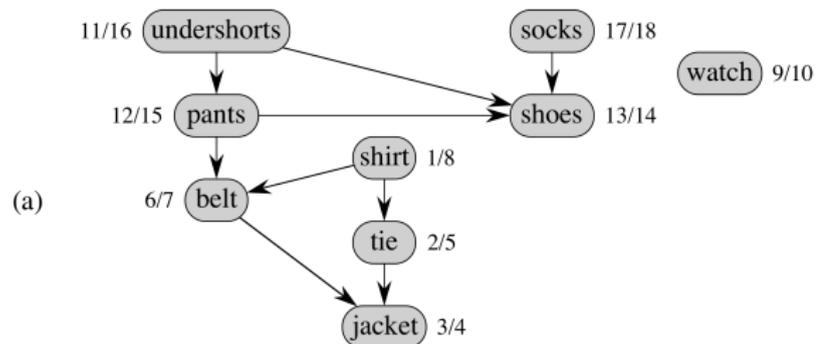
1 chamar DFS(G) para calcular o tempo de término $v.f$ para cada vértice v

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

3 devolver a lista ligada de vértices

Exemplo de execução

Exemplo de execução



Análise do tempo de execução do topological-sort

Análise do tempo de execução do topological-sort

- ▶ O tempo de execução da busca em profundidade é $\Theta(V + E)$
- ▶ O tempo para inserir cada vértice na lista de saída é $O(1)$, cada vértice é inserido apenas uma vez e portanto o tempo total gasto em operações de inserções é de $\Theta(V)$

Análise do tempo de execução do topological-sort

- ▶ O tempo de execução da busca em profundidade é $\Theta(V + E)$
- ▶ O tempo para inserir cada vértice na lista de saída é $O(1)$, cada vértice é inserido apenas uma vez e portanto o tempo total gasto em operações de inserções é de $\Theta(V)$
- ▶ Portanto, o tempo de execução do algoritmo é $\Theta(V + E)$

Corretude do topological-sort

Corretude do topological-sort

- ▶ Precisamos mostrar que se $(u, v) \in E$, então $v.f < u.f$
- ▶ Quando a aresta (u, v) é explorada, quais são as cores de u e v ?

Corretude do topological-sort

- ▶ Precisamos mostrar que se $(u, v) \in E$, então $v.f < u.f$
- ▶ Quando a aresta (u, v) é explorada, quais são as cores de u e v ?
- ▶ u é cinza

Corretude do topological-sort

- ▶ Precisamos mostrar que se $(u, v) \in E$, então $v.f < u.f$
- ▶ Quando a aresta (u, v) é explorada, quais são as cores de u e v ?
- ▶ u é cinza
- ▶ v é cinza também?

Corretude do topological-sort

- ▶ Precisamos mostrar que se $(u, v) \in E$, então $v.f < u.f$
- ▶ Quando a aresta (u, v) é explorada, quais são as cores de u e v ?
- ▶ u é cinza
- ▶ v é cinza também?
 - ▶ Não, porque isto implicaria que v é ancestral de u , e portando a aresta (u, v) seria uma aresta de retorno. Graos não contém arestas de retorno

Corretude do topological-sort

- ▶ Precisamos mostrar que se $(u, v) \in E$, então $v.f < u.f$
- ▶ Quando a aresta (u, v) é explorada, quais são as cores de u e v ?
- ▶ u é cinza
- ▶ v é cinza também?
 - ▶ Não, porque isto implicaria que v é ancestral de u , e portando a aresta (u, v) seria uma aresta de retorno. Graos não contém arestas de retorno
- ▶ v é branco?

Corretude do topological-sort

- ▶ Precisamos mostrar que se $(u, v) \in E$, então $v.f < u.f$
- ▶ Quando a aresta (u, v) é explorada, quais são as cores de u e v ?
- ▶ u é cinza
- ▶ v é cinza também?
 - ▶ Não, porque isto implicaria que v é ancestral de u , e portando a aresta (u, v) seria uma aresta de retorno. Graos não contém arestas de retorno
- ▶ v é branco?
 - ▶ Então v torna-se um descendente de u . Pelo teorema do parênteses $u.d < v.d < \mathbf{v.f} < \mathbf{u.f}$

Corretude do topological-sort

- ▶ Precisamos mostrar que se $(u, v) \in E$, então $v.f < u.f$
- ▶ Quando a aresta (u, v) é explorada, quais são as cores de u e v ?
- ▶ u é cinza
- ▶ v é cinza também?
 - ▶ Não, porque isto implicaria que v é ancestral de u , e portando a aresta (u, v) seria uma aresta de retorno. Graos não contém arestas de retorno
- ▶ v é branco?
 - ▶ Então v torna-se um descendente de u . Pelo teorema do parênteses $u.d < v.d < \mathbf{v.f} < \mathbf{u.f}$
- ▶ v é preto?

Corretude do topological-sort

- ▶ Precisamos mostrar que se $(u, v) \in E$, então $v.f < u.f$
- ▶ Quando a aresta (u, v) é explorada, quais são as cores de u e v ?
- ▶ u é cinza
- ▶ v é cinza também?
 - ▶ Não, porque isto implicaria que v é ancestral de u , e portando a aresta (u, v) seria uma aresta de retorno. Graos não contém arestas de retorno
- ▶ v é branco?
 - ▶ Então v torna-se um descendente de u . Pelo teorema do parênteses $u.d < v.d < \mathbf{v.f} < \mathbf{u.f}$
- ▶ v é preto?
 - ▶ Então v já foi finalizado. Como a aresta (u, v) está sendo explorada, u não foi finalizado. Logo $v.f < u.f$

Referências

Referências

- ▶ Thomas H. Cormen et al. Introduction to Algorithms. 3rd edition. Capítulo 22.4.