Componentes fortemente conexos

Algoritmos em Grafos

Marco A L Barbosa

Conteúdo

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

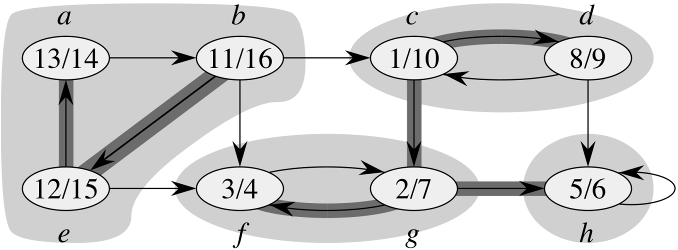
Introdução

- Um **componente fortemente conexo** (SCC) de um grafo orientado $G = (V, E)$ é um conjunto máximo de vértices $C \subseteq V$, tal que, para todo par de vértice u e v
  - $u \rightsquigarrow v$ e $v \rightsquigarrow u$

Introdução

- Um **componente fortemente conexo** (SCC) de um grafo orientado $G = (V, E)$ é um conjunto máximo de vértices $C \subseteq V$, tal que, para todo par de vértice u e v
  - $u \rightsquigarrow v$ e $v \rightsquigarrow u$

(a)



Grafo transposto

- O algoritmo para identificar componentes fortemente conexos utiliza o grafo transposto de G
  - $G^T = (V, E^T), E^T = \{(u, v) : (v, u) \in E\}$
  - G^T é G com todas as arestas invertidas
  - G^T pode ser calculado em tempo $\Theta(V + E)$ para a representação de lista de adjacências
  - G e G^T tem os mesmos SCC's
  - Veja o exercício 22.1-3

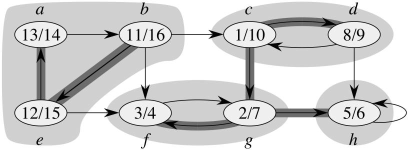
Grafo de componentes

- Grafo de componentes
  - $G^{\text{SCC}} = (V^{\text{SCC}}, E^{\text{SCC}})$
  - V^{SCC} tem um vértice para cada SCC em G
  - E^{SCC} contém uma aresta se existe uma aresta correspondente entre os SCC's de G

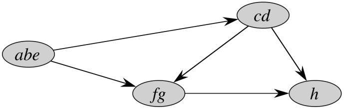
Grafo de componentes

- Grafo de componentes
  - $G^{\text{SCC}} = (V^{\text{SCC}}, E^{\text{SCC}})$
  - V^{SCC} tem um vértice para cada SCC em G
  - E^{SCC} contém uma aresta se existe uma aresta correspondente entre os SCC's de G

(a)



(c)



Grafo de componentes

- Lema 22.13
  - G^{SCC} é um gao
  - Sejam C e C' SCC distintos em G, seja $u, v \in C$ e seja $u', v' \in C'$. Suponha que exista um caminho $u \rightsquigarrow u'$ em G. Então, não pode existir um caminho $v' \rightsquigarrow v$ em G

Grafo de componentes

- Lema 22.13
  - G^{SCC} é um gao
  - Sejam C e C' SCC distintos em G, seja $u, v \in C$ e seja $u', v' \in C'$. Suponha que exista um caminho $u \leadsto u'$ em G. Então, não pode existir um caminho $v' \leadsto v$ em G
  - Prova: Suponha que exista um caminho $v' \leadsto v$ em G. Então existem caminhos $u \leadsto u' \leadsto v'$ e $v' \leadsto v \leadsto u$ em G. Portanto, u e v' são acessíveis um a partir do outro, e não podem estar em SCC separados

Procedimento

`strongly-connected-components`

Procedimento `strongly-connected-components`

`strongly-connected-components(G)`

1 chamar `dfs(G)` para calcular o tempo de término $u.f$ para cada vértice u

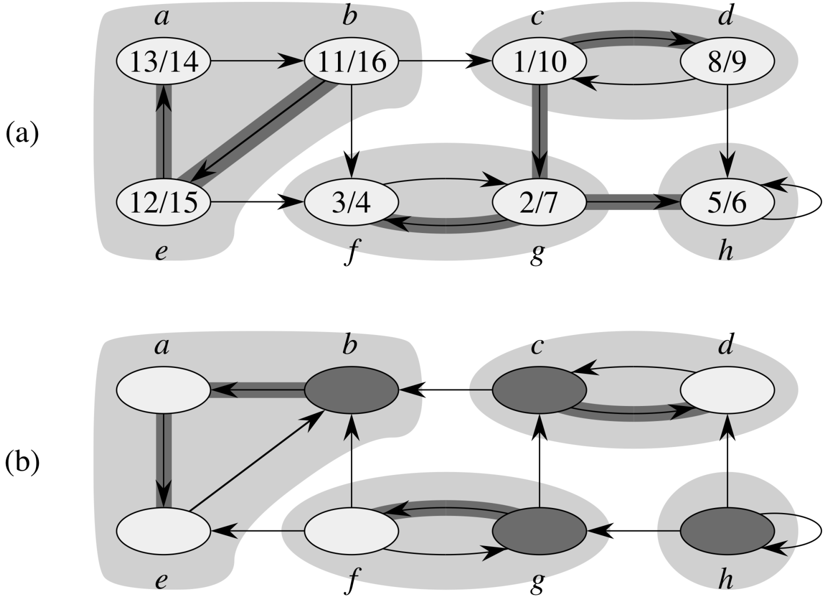
2 calcular G^T

3 chamar `dfs(`G^T`)` mas, no laço principal de `dfs`, considerar os vértices em ordem decrescente de $u.f$

4 os vértices de cada árvore na floresta primeiro na profundidade formada na linha 3 formam um componente fortemente conexo

Exemplo de execução

Exemplo de execução



Análise do tempo de execução do `strongly-connected-components`

Análise do tempo de execução do `strongly-connected-components`

- O tempo do `dfs` das linhas 1 e 3 é $\Theta(V + E)$
- Conforme os vértices são terminados na chamada do `dfs` da linha 1, os vértices são inseridos na frente de uma lista ligada ($O(1)$), como cada vértice é inserido apenas uma vez, o tempo total de operações de inserções é $\Theta(V)$
- O tempo para calcular o grafo transposto na linha 2 é $\Theta(V + E)$
- Portanto, o tempo de execução do algoritmo é $\Theta(V + E)$

Corretude do `strongly-connected-components`

Corretude do `strongly-connected-components`

- Ideia
  - Considerando os vértices no segundo `dfs` na ordem decrescente dos tempos de términos obtidos no primeiro `dfs`, estamos visitando os vértices do grafo de componentes na ordem topológica

Corretude do `strongly-connected-components`

- Vamos definir duas questões de notação
  - As referências a $u.d$ e $u.f$ referem-se aos valores do primeiro `dfs`
  - Para um conjunto $U \subseteq V$, definimos
    - $d(U) = \min_{u \in U}\{u.d\}$ (tempo de descoberta mais antigo)
    - $f(U) = \max_{u \in U}\{u.f\}$ (tempo de término mais recente)

Corretude do `strongly-connected-components`

- Lema 22.14
  - Sejam C e C' SCC distintos em $G = (V, E)$. Suponha que exista uma aresta $(u, v) \in E$, tal que $u \in C$ e $v \in C'$. Então $f(C) > f(C')$

Corretude do `strongly-connected-components`

- Lema 22.14
  - Sejam C e C' SCC distintos em $G = (V, E)$. Suponha que exista uma aresta $(u, v) \in E$, tal que $u \in C$ e $v \in C'$. Então $f(C) > f(C')$
- Corolário 22.15
  - Sejam C e C' SCC distintos em $G = (V, E)$. Suponha que exista uma aresta $(u, v) \in E^T$, tal que $u \in C$ e $v \in C'$. Então $f(C) < f(C')$

Corretude do `strongly-connected-components`

- Teorema 22.16: `strongly-connected-components(G)` calcula corretamente os SCC's de um grafo orientado G
  - O segundo `dfs` começa com um SCC C tal que $f(C)$ é máximo
  - Seja $x \in C$ o vértice inicial, o segundo `dfs` visita todos os vértices de C. Pelo corolário, como $f(C) > f(C')$ para todo $C \neq C'$, não existe aresta de C para C'. Logo, o `dfs` visita apenas os vértices de C (descobrindo este SCC)
  - A próxima raiz escolhida no segundo `dfs` está em um SCC C' tal que $f(C')$ é máximo em relação a todos os outros SCC (sem considerar C). O `dfs` visita todos os vértices de C', e as únicas arestas fora de C' vão para C, cujo os vértices já foram visitados
  - O processo continua até que todos os vértices sejam visitados

Corretude do `strongly-connected-components`

- Teorema 22.16: `strongly-connected-components(G)` calcula corretamente os SCC de um grafo orientado G
  - Cada vez que uma raiz é escolhida pelo segundo `dfs`, ele só pode alcançar
    - Os vértices no SCC dele (através de arestas da árvore)
    - Os vértices que já foram visitados no segundo `dfs`

Referências

Referências

- Thomas H. Cormen et al. Introduction to Algorithms. 3rd edition. Capítulo 22.5.