

Componentes fortemente conexas

CLRS Cap 22.5

Busca em profundidade

DFS (G)

- 1 **para cada** $u \in V(G)$ **faça**
- 2 $u.cor \leftarrow$ branco $u.\pi \leftarrow$ nil
- 3 tempo $\leftarrow 0$
- 4 **para cada** $u \in V(G)$ **faça**
- 5 **se** $u.cor =$ branco
- 6 **então** DFS-Visit(u)

Busca em profundidade

DFS (G)

- 1 **para cada** $u \in V(G)$ **faça**
- 2 $u.cor \leftarrow$ branco $u.\pi \leftarrow$ nil
- 3 $tempo \leftarrow 0$
- 4 **para cada** $u \in V(G)$ **faça**
- 5 **se** $u.cor =$ branco
- 6 **então** DFS-Visit(u)

DFS-Visit(u)

- 1 $u.cor \leftarrow$ cinzento $u.d \leftarrow$ tempo $tempo \leftarrow$ tempo + 1
- 3 **para cada** $v \in u.Estrela$ **faça**
- 4 **se** $u.cor =$ branco
- 5 **então** $v.\pi \leftarrow u$
- 6 DFS-Visit(v)
- 7 $u.cor \leftarrow$ preto
- 8 $u.f \leftarrow$ tempo $tempo \leftarrow$ tempo + 1

A floresta DF

Cada componente da floresta DF é uma árvore enraizada, orientada a partir da raiz.

A floresta DF

Cada componente da floresta DF é uma árvore enraizada, orientada a partir da raiz.

A cada chamada de DFS-Visit no laço principal, os vértices lá processados fazem parte da componente cuja raiz é o argumento da chamada.

A floresta DF

Cada componente da floresta DF é uma árvore enraizada, orientada a partir da raiz.

A cada chamada de DFS-Visit no laço principal, os vértices lá processados fazem parte da componente cuja raiz é o argumento da chamada.

Extensão do algoritmo para capturar essas componentes...

A floresta DF

Cada componente da floresta DF é uma árvore enraizada, orientada a partir da raiz.

A cada chamada de DFS-Visit no laço principal, os vértices lá processados fazem parte da componente cuja raiz é o argumento da chamada.

Extensão do algoritmo para capturar essas componentes...

Sugestões?

Componentes fortemente conexas

Seja G um digrafo e u e v vértices de G .

Componentes fortemente conexas

Seja G um digrafo e u e v vértices de G .

Escrevemos $u \rightsquigarrow v$ se existe caminho de u para v em G .

Componentes fortemente conexas

Seja G um digrafo e u e v vértices de G .

Escrevemos $u \rightsquigarrow v$ se existe caminho de u para v em G .

Seja C um conjunto maximal de vértices de G tal que $u \rightsquigarrow v$ e $v \rightsquigarrow u$ quaisquer que sejam u e v em C .

Componentes fortemente conexas

Seja G um digrafo e u e v vértices de G .

Escrevemos $u \rightsquigarrow v$ se existe caminho de u para v em G .

Seja C um conjunto maximal de vértices de G tal que $u \rightsquigarrow v$ e $v \rightsquigarrow u$ quaisquer que sejam u e v em C .

C é uma **componente fortemente conexa** de G .

Componentes fortemente conexas

Seja G um digrafo e u e v vértices de G .

Escrevemos $u \rightsquigarrow v$ se existe caminho de u para v em G .

Seja C um conjunto maximal de vértices de G tal que $u \rightsquigarrow v$ e $v \rightsquigarrow u$ quaisquer que sejam u e v em C .

C é uma **componente fortemente conexa** de G .

Diferentes componentes fortemente conexas são disjuntas, ou seja, as componentes determinam uma partição de V_G .

Componentes fortemente conexas

Seja G um digrafo e u e v vértices de G .

Escrevemos $u \rightsquigarrow v$ se existe caminho de u para v em G .

Seja C um conjunto maximal de vértices de G tal que $u \rightsquigarrow v$ e $v \rightsquigarrow u$ quaisquer que sejam u e v em C .

C é uma **componente fortemente conexa** de G .

Diferentes componentes fortemente conexas são disjuntas, ou seja, as componentes determinam uma partição de V_G .

Problema: Dado digrafo G , encontrar todas as componentes fortemente conexas de G .

Algoritmo de Kosaraju-Sharir

Seja $G = (V, E)$ um digrafo.

Seja G^r o digrafo reverso de G (todos os arcos invertidos).

Algoritmo de Kosaraju-Sharir

Seja $G = (V, E)$ um digrafo.

Seja G^r o digrafo reverso de G (todos os arcos invertidos).

Kosaraju-Sharir(G)

Algoritmo de Kosaraju-Sharir

Seja $G = (V, E)$ um digrafo.

Seja G^r o digrafo reverso de G (todos os arcos invertidos).

Kosaraju-Sharir(G)

Execute uma DFS em G calculando $u.f$ para cada u .

Algoritmo de Kosaraju-Sharir

Seja $G = (V, E)$ um digrafo.

Seja G^r o digrafo reverso de G (todos os arcos invertidos).

Kosaraju-Sharir(G)

Execute uma DFS em G calculando $u.f$ para cada u .

Construa G^r .

Algoritmo de Kosaraju-Sharir

Seja $G = (V, E)$ um digrafo.

Seja G^r o digrafo reverso de G (todos os arcos invertidos).

Kosaraju-Sharir(G)

Execute uma DFS em G calculando $u.f$ para cada u .

Construa G^r .

Execute uma DFS em G^r considerando os vértices de G^r em ordem decrescente do valor de f calculado acima.

Algoritmo de Kosaraju-Sharir

Seja $G = (V, E)$ um digrafo.

Seja G^r o digrafo reverso de G (todos os arcos invertidos).

Kosaraju-Sharir(G)

Execute uma DFS em G calculando $u.f$ para cada u .

Construa G^r .

Execute uma DFS em G^r considerando os vértices de G^r em ordem decrescente do valor de f calculado acima.

Devolva as componentes da floresta DF construída para G^r .

Algoritmo de Kosaraju-Sharir

Seja $G = (V, E)$ um digrafo.

Seja G^r o digrafo reverso de G (todos os arcos invertidos).

Kosaraju-Sharir(G)

Execute uma DFS em G calculando $u.f$ para cada u .

Construa G^r .

Execute uma DFS em G^r considerando os vértices de G^r em ordem decrescente do valor de f calculado acima.

Devolva as componentes da floresta DF construída para G^r .

Algoritmo de Kosaraju-Sharir

Seja $G = (V, E)$ um digrafo.

Seja G^r o digrafo reverso de G (todos os arcos invertidos).

Kosaraju-Sharir(G)

Execute uma DFS em G calculando $u.f$ para cada u .

Construa G^r .

Execute uma DFS em G^r considerando os vértices de G^r em ordem decrescente do valor de f calculado acima.

Devolva as componentes da floresta DF construída para G^r .

Consumo de tempo: linear no tamanho de G .

Análise

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G e vértices $u, v \in C$ e $u', v' \in C'$. Se existe um caminho de u a u' em G , então não existe caminho em G de v' a v .

Análise

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G e vértices $u, v \in C$ e $u', v' \in C'$. Se existe um caminho de u a u' em G , então não existe caminho em G de v' a v .

d e f como calculados na primeira DFS do algoritmo.

Análise

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G e vértices $u, v \in C$ e $u', v' \in C'$. Se existe um caminho de u a u' em G , então não existe caminho em G de v' a v .

d e f como calculados na primeira DFS do algoritmo.

Para um conjunto U de vértices,

sejam $d(U) := \min\{u.d : u \in U\}$

e $f(U) := \max\{u.f : u \in U\}$.

Análise

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G e vértices $u, v \in C$ e $u', v' \in C'$. Se existe um caminho de u a u' em G , então não existe caminho em G de v' a v .

d e f como calculados na primeira DFS do algoritmo.

Para um conjunto U de vértices,

sejam $d(U) := \min\{u.d : u \in U\}$

e $f(U) := \max\{u.f : u \in U\}$.

$d(U)$: instante em que o primeiro vértice de U foi descoberto

Análise

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G e vértices $u, v \in C$ e $u', v' \in C'$. Se existe um caminho de u a u' em G , então não existe caminho em G de v' a v .

d e f como calculados na primeira DFS do algoritmo.

Para um conjunto U de vértices,

sejam $d(U) := \min\{u.d : u \in U\}$

e $f(U) := \max\{u.f : u \in U\}$.

$d(U)$: instante em que o primeiro vértice de U foi descoberto

$f(U)$: instante em que o último vértice de U terminou

Análise

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G e vértices $u, v \in C$ e $u', v' \in C'$. Se existe um caminho de u a u' em G , então não existe caminho em G de v' a v .

d e f como calculados na primeira DFS do algoritmo.

Para um conjunto U de vértices,

sejam $d(U) := \min\{u.d : u \in U\}$

e $f(U) := \max\{u.f : u \in U\}$.

$d(U)$: instante em que o primeiro vértice de U foi descoberto

$f(U)$: instante em que o último vértice de U terminou

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G . Se existe um arco (u, u') em G , com $u \in C$ e $u' \in C'$, então $f(C) > f(C')$.

Análise

d e f como calculados na primeira DFS do algoritmo.

Para um conjunto U de vértices,

seja $d(U) := \min\{u.d : u \in U\}$ e $f(U) := \max\{u.f : u \in U\}$.

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G . Se existe um arco (u, u') em G^r , então $f(C) < f(C')$.

Análise

d e f como calculados na primeira DFS do algoritmo.

Para um conjunto U de vértices,

seja $d(U) := \min\{u.d : u \in U\}$ e $f(U) := \max\{u.f : u \in U\}$.

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G . Se existe um arco (u, u') em G^r , então $f(C) < f(C')$.

Isso implica que a ordenação por f decrescente na segunda DFS percorre as componentes fortemente conexas.

Análise

d e f como calculados na primeira DFS do algoritmo.

Para um conjunto U de vértices,

seja $d(U) := \min\{u.d : u \in U\}$ e $f(U) := \max\{u.f : u \in U\}$.

Lema: Sejam C e C' componentes fortemente conexas distintas de um digrafo G . Se existe um arco (u, u') em G^r , então $f(C) < f(C')$.

Isso implica que a ordenação por f decrescente na segunda DFS percorre as componentes fortemente conexas.

Teorema: Kosaraju-Sharir(G) calcula corretamente as componentes fortemente conexas de G .

Condensação de um digrafo

Dado um digrafo G , sua **condensação** é o digrafo cujos vértices são (ou estão em correspondência com) as componentes conexas de G , com arco de C a C' sse existe em G arco de um vértice em C a um de C' .

TEOREMA: *A condensação de um digrafo é um digrafo acíclico.*

Condensação de um digrafo

Dado um digrafo G , sua **condensação** é o digrafo cujos vértices são (ou estão em correspondência com) as componentes conexas de G , com arco de C a C' sse existe em G arco de um vértice em C a um de C' .

TEOREMA: *A condensação de um digrafo é um digrafo acíclico.*

PROBLEMA: Adaptar [Kosaraju-Sharir](#) para devolver uma ordenação topológica da condensação de G .