

Graphen I

Gustavo Crivelli, Beini Ma, Matthias Schimek, Matthias Schmitt

ICPC-Praktikum 2015 - Graphen I

Einleitung

BFS

DFS

Iterative Tiefensuche

Starke Zusammenhangskomponenten

bipartite Graphen

Brücke

SCCs finden mittels DFS

Beispielaufgabe

UVa 11383 Come and Go

In einer Stadt gibt es N Kreuzungen, die durch Straßen verbunden sind. Da man in der Stadt von einem Punkt (Kreuzung) zu jedem anderen kommen möchte, sollte es eine Verbindung zwischen zwei beliebigen Kreuzungen geben.

Für eine gegebene Stadt mit N Kreuzungen und M Straßen soll entschieden werden, ob dies möglich ist.

Definition Strongly Connected Components SCC

In einem gerichteten Graph $G = (V, E)$, wird $V' \subseteq V$ starke Zusammenhangskomponente (SCC) genannt, wenn zwischen je zwei Knoten in V' ein Pfad existiert.

SCCs finden mittels DFS

Beispielaufgabe

UVa 11383 Come and Go

In einer Stadt gibt es N Kreuzungen, die durch Straßen verbunden sind. Da man in der Stadt von einem Punkt (Kreuzung) zu jedem anderen kommen möchte, sollte es eine Verbindung zwischen zwei beliebigen Kreuzungen geben.

Für eine gegebene Stadt mit N Kreuzungen und M Straßen soll entschieden werden, ob dies möglich ist.

Definition Strongly Connected Components SCC

In einem gerichteten Graph $G = (V, E)$, wird $V' \subseteq V$ starke Zusammenhangskomponente (SCC) genannt, wenn zwischen je zwei Knoten in V' ein Pfad existiert.

- Zum Lösen der Aufgabe untersuchen, ob das Straßennetz der Stadt aus einer oder mehreren SCCs besteht.
- \Rightarrow Benötigen effizienten Algorithmus zum Finden von SCCs

Algorithmus von Tarjan für SCCs

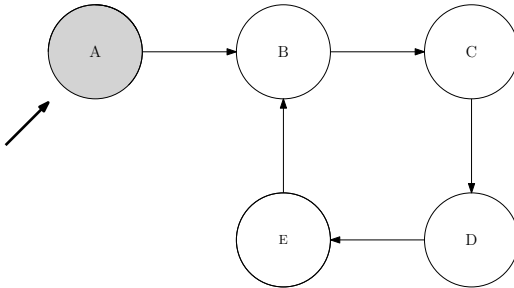
- Wurde von Robert Tarjan gefunden
- Basiert auf dem Konzept der DFS
- Laufzeit: $\mathcal{O}(|V| + |E|)$

- Zum Lösen der Aufgabe untersuchen, ob das Straßennetz der Stadt aus einer oder mehreren SCCs besteht.
- \Rightarrow Benötigen effizienten Algorithmus zum Finden von SCCs

Algorithmus von Tarjan für SCCs

- Wurde von Robert Tarjan gefunden
- Basiert auf dem Konzept der DFS
- Laufzeit: $\mathcal{O}(|V| + |E|)$

-
- $\text{dfs num}(A) = 0$
 - $\text{dfs low}(A) = 0$



STACK:

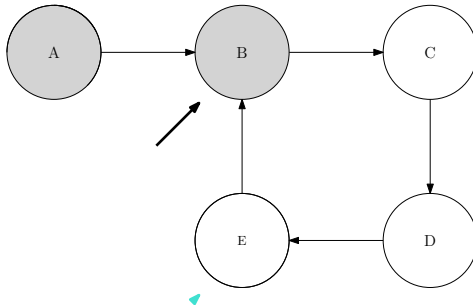
- A

-
- $\text{dfs num}(A) = 0$
 - $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

STACK:

- A
- B



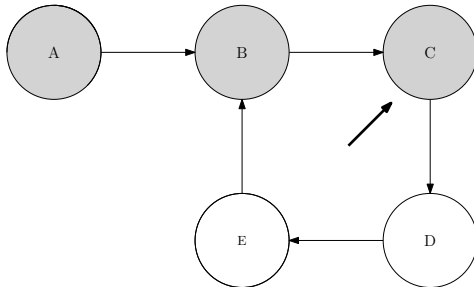
-
- $\text{dfs num}(A) = 0$
 - $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 2$

STACK:

- A
- B
- C



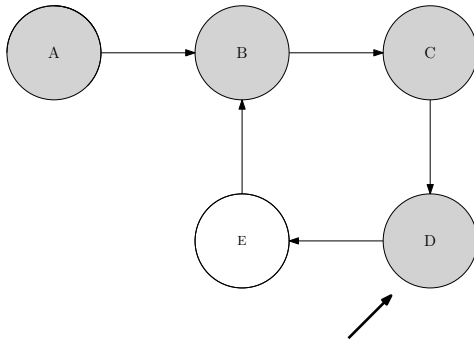
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 2$

STACK:

- A
- B
- C
- D



- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 3$

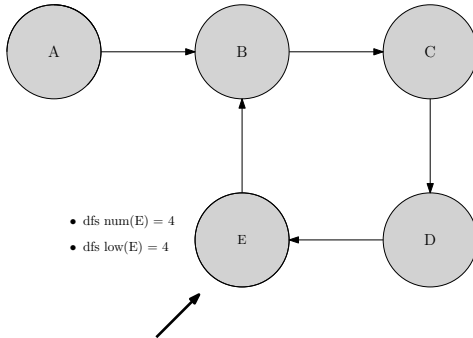
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 2$

STACK:

- A
- B
- C
- D
- E



- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 4$

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 3$

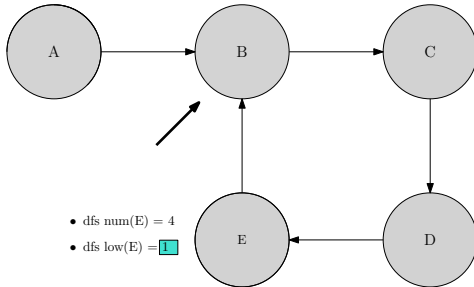
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 2$

STACK:

- A
- B
- C
- D
- E



- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 1$

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 3$

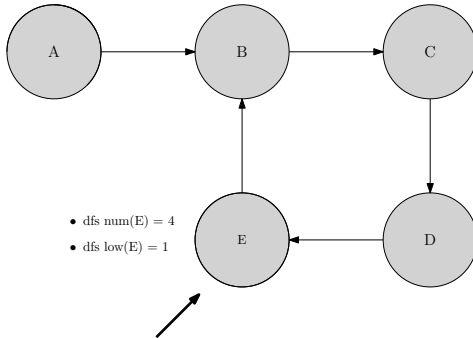
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 2$

STACK:

- A
- B
- C
- D
- E



- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 1$

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 1$

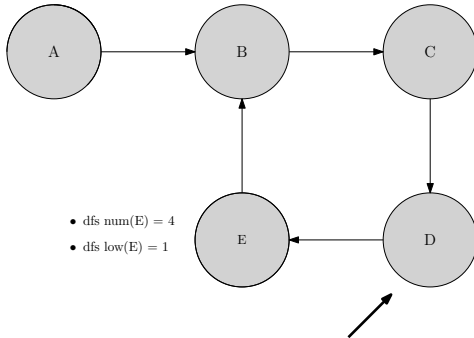
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 1$

STACK:

- A
- B
- C
- D
- E



- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 1$

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 1$

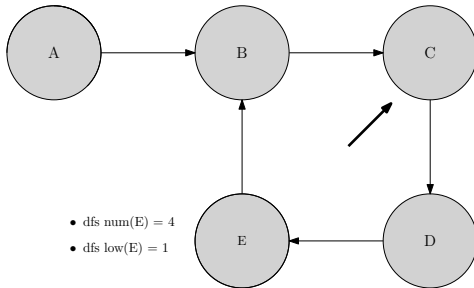
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 1$

STACK:

- A
- B
- C
- D
- E



- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 1$

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 1$

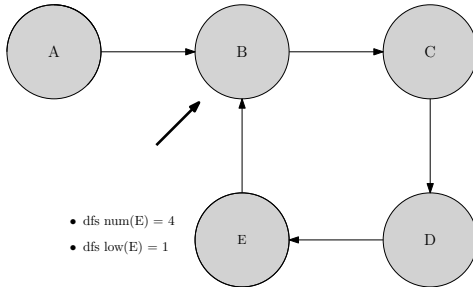
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 1$

STACK:

- A
- B
- C
- D



- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 1$

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 1$

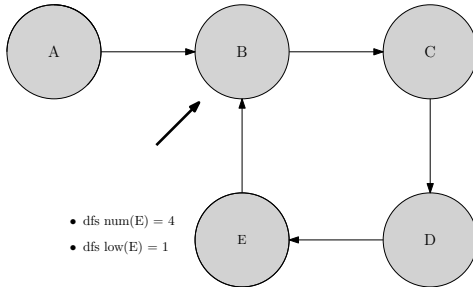
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 1$

STACK:

- A
- B
- C



- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 1$

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 1$

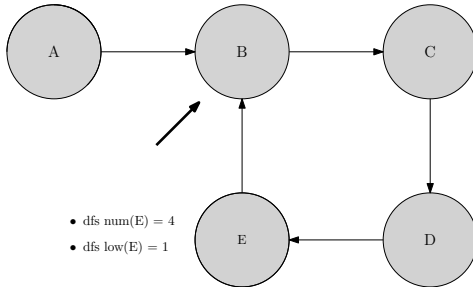
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$

- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = 1$

- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 1$

STACK:

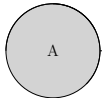
- A
- B



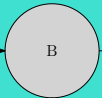
- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 1$

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 1$

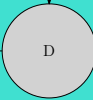
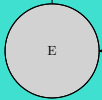
- $\text{dfs num}(A) = 0$
- $\text{dfs low}(A) = 0$



- $\text{dfs num}(B) = 1$
- $\text{dfs low}(B) = \boxed{1}$
- $\text{dfs num}(C) = 2$
- $\text{dfs low}(C) = 1$



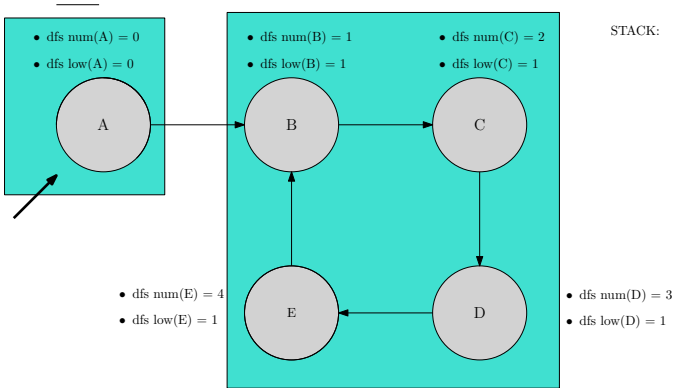
- $\text{dfs num}(E) = 4$
- $\text{dfs low}(E) = 1$



STACK:

- A

- $\text{dfs num}(D) = 3$
- $\text{dfs low}(D) = 1$



Ende