



NTNU

Algoritmer og Datastrukturer
~~Innovation and Creativity~~

Topologisk sortering, SCC og minimale spenntreer. (intro)

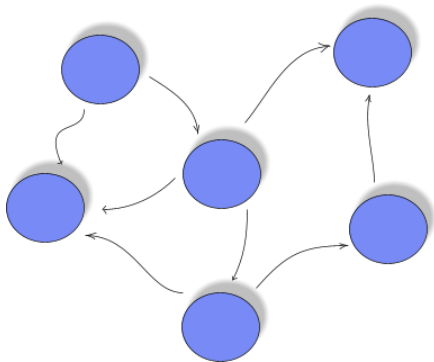
Simon Jonassen

Institutt for Datateknikk og Informasjonsvitenskap

September 12, 2006

Topologisk sortering

DAG (directed acyclic graph) - rettet asyklisk graf.
En rettet graf er asyklisk hvis og bare hvis den har ingen tilbakeførende kanter (back edges).



Topologisk Sortering

Problemstilling

Finner en ordning av alle noder i en DAG slik at alle noder som starter en kant listes opp før alle noder som avslutter den.

Algoritme

- Kjør DFS på alle noder og legger dem i front av en liste når vi blir ferdige med dem.

Strongly Connected Components

Definisjon

SCC av en graf $G = (V, E)$ er en maksimal mengde av alle hjørner, $C \subseteq E$, slik at for hvert par av hjørner $u, v \in C$ har vi både $u \rightsquigarrow v$ og $v \rightsquigarrow u$ (dvs. begge hjørner er oppnåelige fra hverandre).

Algoritme

- Kjør DFS på G og finn alle endetider på G .
- Kjør DFS på G^T men start etter synkende endetid
- Resulterende dybde først skog separerer SCC'er.

$$G^T = (V, E^T) \text{ der } E^T = \{(u, v) \mid (v, u) \in E\}$$

Minimale spenntrær

Gitt en graf $G = (V, E)$. Et minimalt spenntrær er en asyklisk delmengde T , av alle elementer $(u, v) \in E$ som sammenkobler alle noder og som har $w(T) = \sum_{(u,v) \in T} w(u, v)$ minimal.

Minimale spenntrær

Forutsetning: for hvert steg er A en delmengde av et MST til G .

Generisk MST

Generic-MST(G, w)

$A \leftarrow \emptyset$

while A does not form a spanning tree

 find an edge (u, v) that is safe for A

$A \leftarrow A \cup \{(u, v)\}$

return A

En kant er trygg hvis den kan legges til uten å bryte forutsetningen over.

Minimale spenntær

Kruskal

MST-Kruskal(G, w)

$A \leftarrow \emptyset$

foreach vertex $v \in V[G]$

$MakeSet(v)$

sort elements of E into nondecreasing order by w .

foreach $(u, v) \in E$

 if $FindSet(v) \neq FindSet(u)$

$A \leftarrow A \cup \{(u, v)\}$

$Union(u, v)$

return A

Minimale spenntær

Prim (aka. Dijkstra-Jarnik-Prim)

```
MST-Prim( $G, w, r$ )
  foreach  $u \in V[G]$ 
     $key[u] \leftarrow inf$ 
     $\pi[u] \leftarrow nil$ 
   $key[r] \leftarrow 0$ 
   $Q \leftarrow V[G]$ 
  while  $Q \neq \emptyset$ 
     $u \leftarrow ExtractMin(Q)$ 
    foreach  $v$  adjacent to  $u$ 
      if  $v \in Q$  and  $w(u, v) < key[v]$ 
         $\pi[v] \leftarrow u$ 
         $key[v] \leftarrow w(u, v)$ 
```