INF2220 - Algoritmer og datastrukturer

HØSTEN 2015

Institutt for informatikk, Universitetet i Oslo

Forelesning 8:

Grafer III

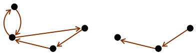
Dagens plan:

Dybde-først søk

- Strongly connected components

Gjesteforelesning: "Grafalgoritmer på verdens største graf "
av Torbjørn Morland

rettet graf & DFS



En rettet graf er sterkt sammenhengende hvis og bare hvis vi fra hver eneste node **v** klarer å besøke alle de andre nodene i grafen ved et dybde-først søk fra **v** partisjonere G i såkalt strongly connected components

Definition (SCC)

Gitt en rettet graf $G = (V, E)$. En *strongly connected component* av G er en maksimal sett av noder $U \subseteq V$ s.t.: for alle $u_1, u_2 \in U$ vi har at $u_1 \rightarrow^* u_2$ and $u_2 \rightarrow^* u_1$.

- Ide: G og G^t har den samme SCC's $\Longrightarrow$ bruk dfs 2 ganger, en gang på G og en gang på den reverserte grafen[1] G^t
- kompleksitet: lineær tid $\mathcal{O}(E + V)$

[1] Gitt $G = (V, E)$ da er $G^t = (V, E^t)$ hvor $(v, u) \in E^t$ iff $(u, v) \in E$

Strongly connected components (SCC)

partisjonere G i såkalt strongly connected components

Definition (SCC)

Gitt en rettet graf $G = (V, E)$. En *strongly connected component* av G er en maksimal sett av noder $U \subseteq V$ s.t.: for alle $u_1, u_2 \in U$ vi har at $u_1 \to^* u_2$ and $u_2 \to^* u_1$.

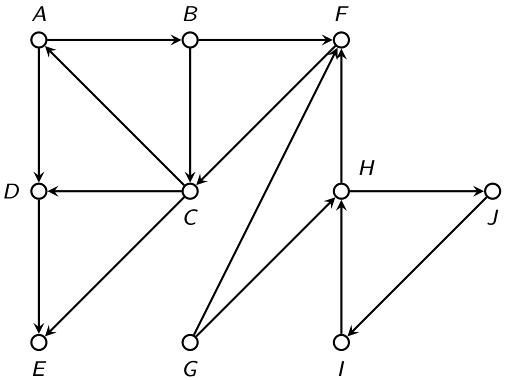
- Ide: G og G^t har den samme SCC's $\Longrightarrow$ bruk dfs 2 ganger, en gang på G og en gang på den reverserte grafen[1] G^t
- kompleksitet: lineær tid $\mathcal{O}(E + V)$

[1] Gitt $G = (V, E)$ da er $G^t = (V, E^t)$ hvor $(v, u) \in E^t$ iff $(u, v) \in E$

SCC

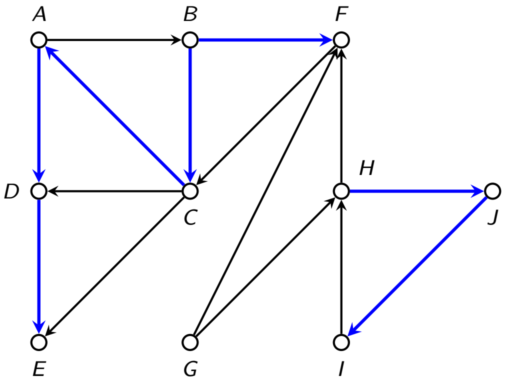
1. DFS på G

1. gjør en DFS traversering
2. husker post-order
(finished time stamp)



1. DFS på G

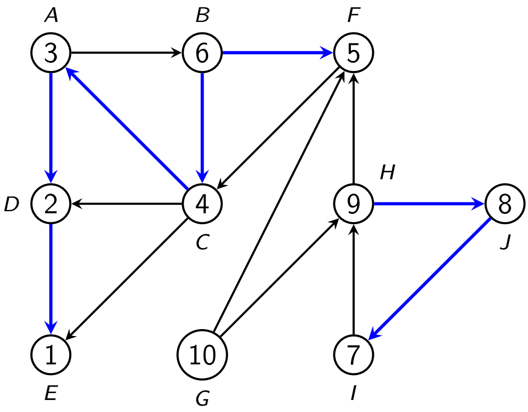
1. gjør en DFS traversering
2. husker post-order (finished time stamp)



SCC

1. DFS på G

1. gjør en DFS traversering
2. husker post-order (`finished` time stamp)



SCC

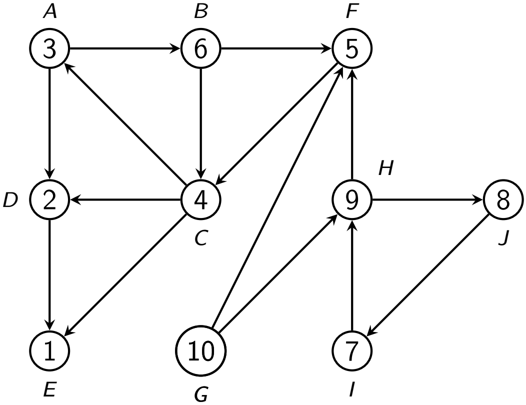
2. Reversere

reversere kantene til G og får G^t

3. DFS på G^t

iterere DFS på G^t i avtagende rekkefølger!

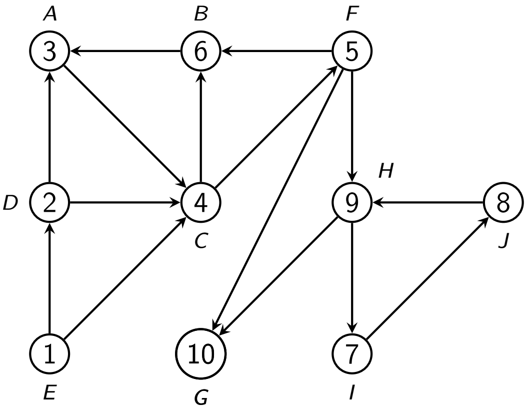
Andre fase (1)



strongly connected components:

{}

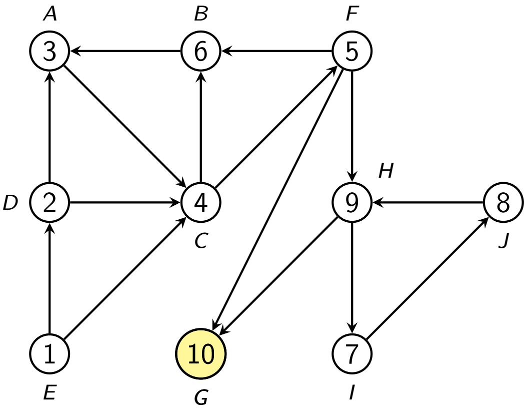
Andre fase (2)



strongly connected components:

{}

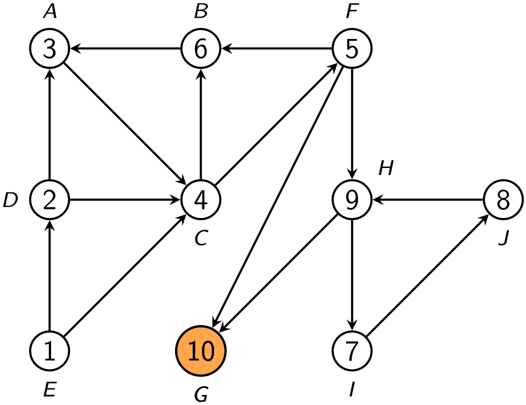
Andre fase (3)



strongly connected components:

$$\{\}$$

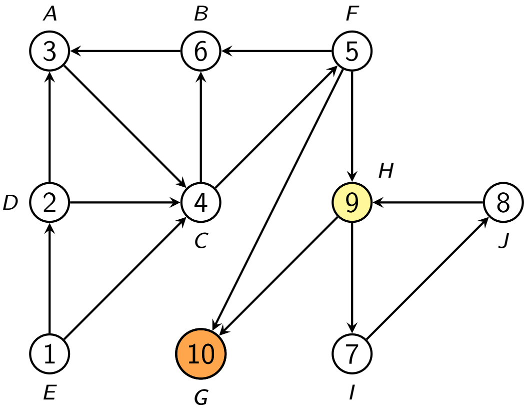
Andre fase (4)



strongly connected components:

$$\{\{G\}\}$$

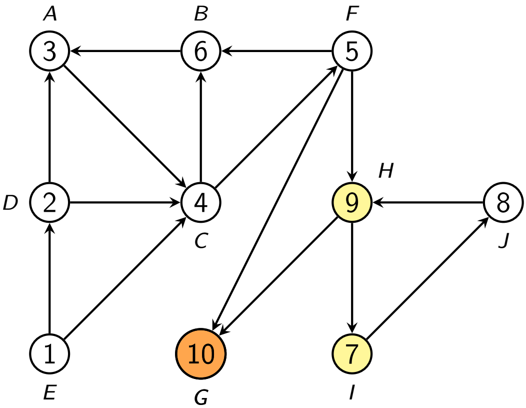
Andre fase (5)



strongly connected components:

$$\{\{G\}\}$$

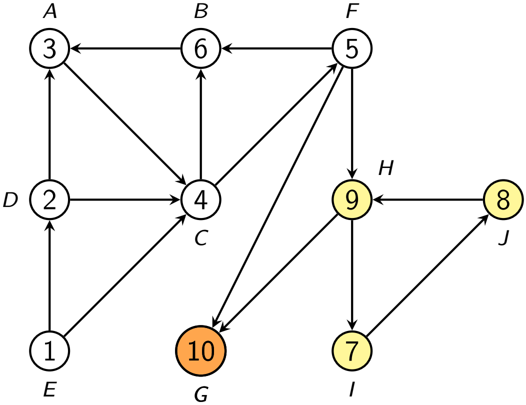
Andre fase (6)



strongly connected components:

$$\{\{G\}\}$$

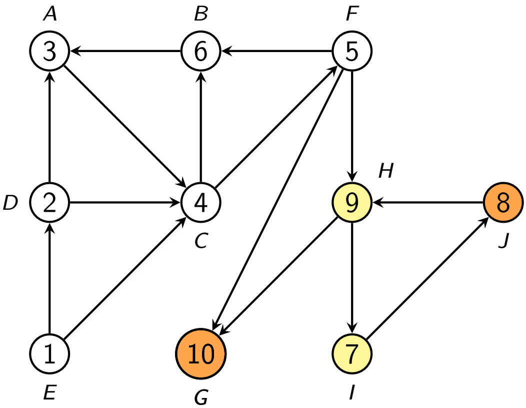
Andre fase (7)



strongly connected components:

$$\{\{G\}\}$$

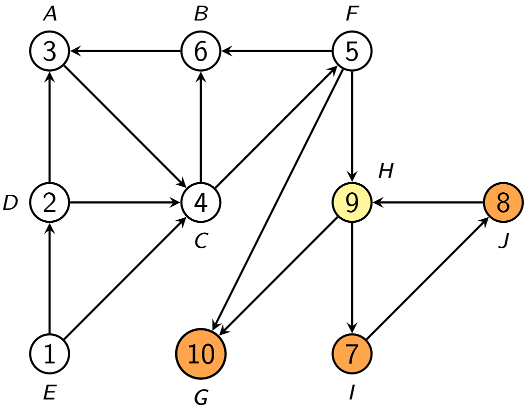
Andre fase (8)



strongly connected components:

$$\{\{G\}\}$$

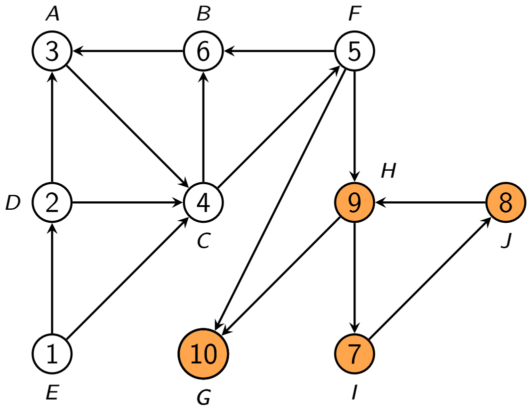
Andre fase (9)



strongly connected components:

$$\{\{G\}\}$$

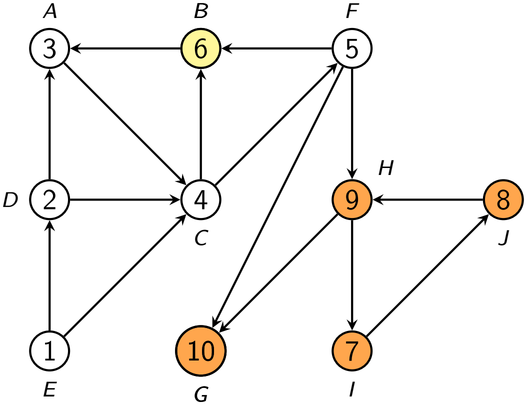
Andre fase (10)



strongly connected components:

$$\{\{G\}, \{H, J, I\}\}$$

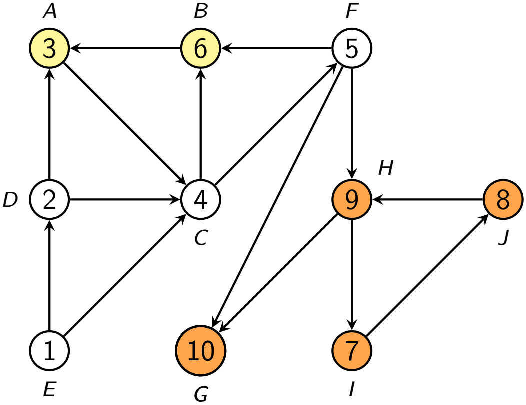
Andre fase (11)



strongly connected components:

$$\{\{G\}, \{H, J, I\}\}$$

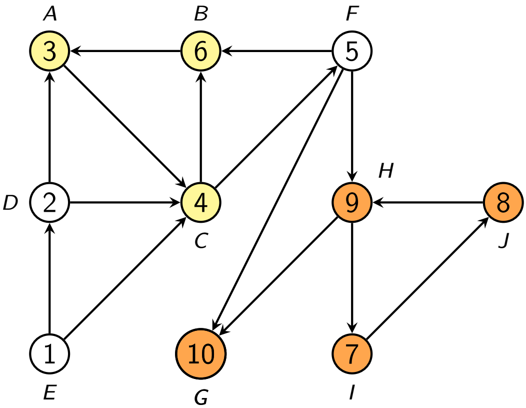
Andre fase (12)



strongly connected components:

$$\{\{G\}, \{H, J, I\}\}$$

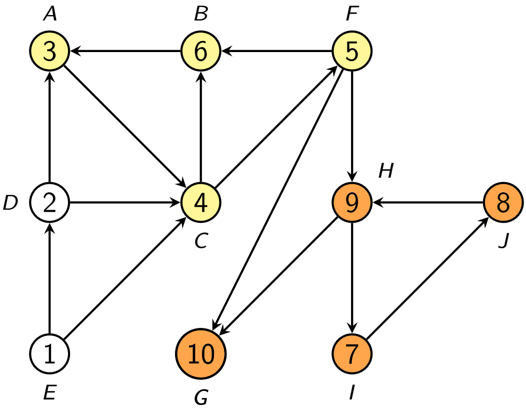
Andre fase (13)



strongly connected components:

$$\{\{G\}, \{H, J, I\}\}$$

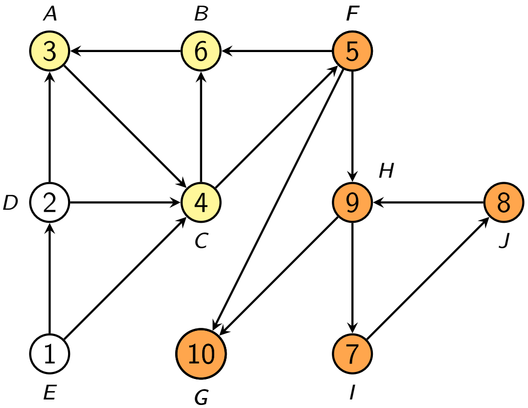
Andre fase (14)



strongly connected components:

$$\{\{G\}, \{H, J, I\}\}$$

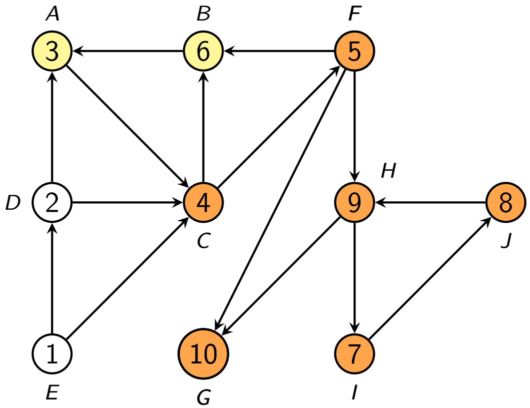
Andre fase (15)



strongly connected components:

$$\{\{G\}, \{H, J, I\}\}$$

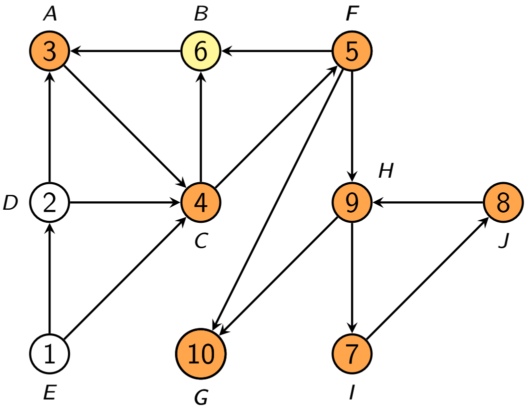
Andre fase (16)



strongly connected components:

$$\{\{G\}, \{H, J, I\}\}$$

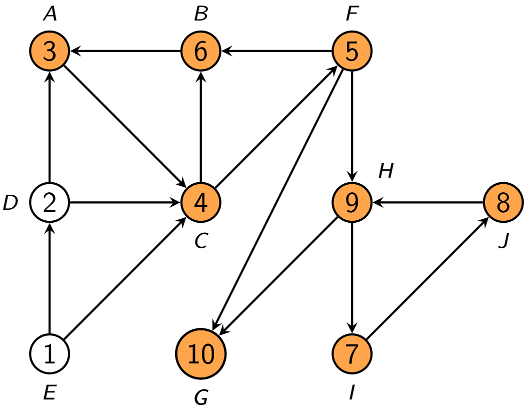
Andre fase (17)



strongly connected components:

$$\{\{G\}, \{H, J, I\}\}$$

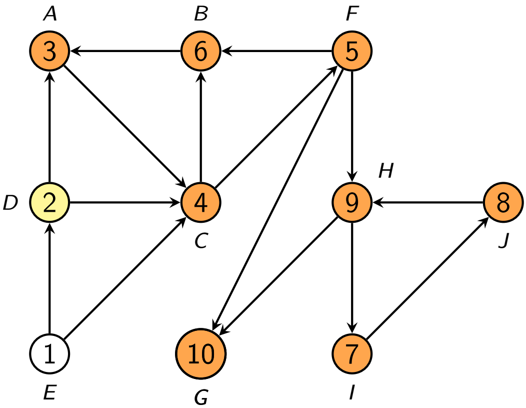
Andre fase (18)



strongly connected components:

$$\{\{G\}, \{H, J, I\}, \{B, A, C, F\}\}$$

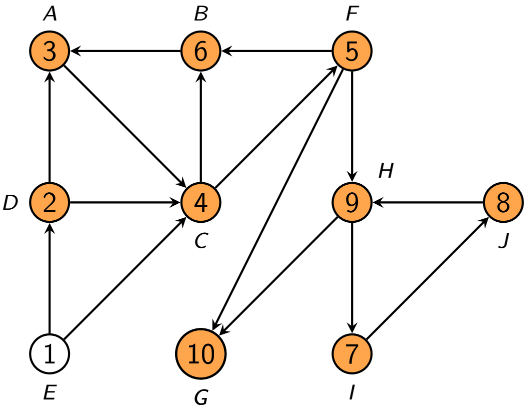
Andre fase (19)



strongly connected components:

$$\{\{G\}, \{H, J, I\}, \{B, A, C, F\}\}$$

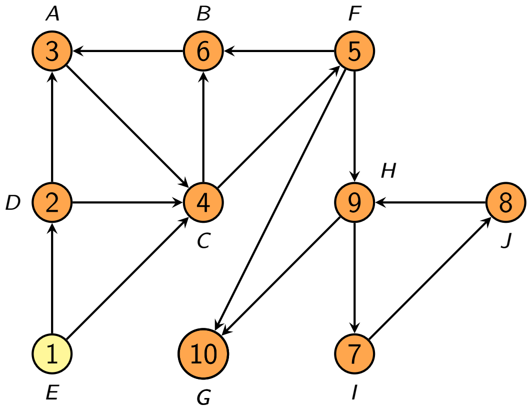
Andre fase (20)



strongly connected components:

$$\{\{G\}, \{H, J, I\}, \{B, A, C, F\}, \{D\}\}$$

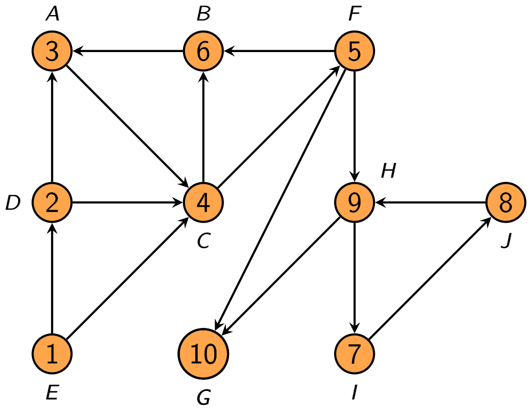
Andre fase (21)



strongly connected components:

$$\{\{G\}, \{H, J, I\}, \{B, A, C, F\}, \{D\}\}$$

Andre fase (22)



strongly connected components:

$$\{\{G\}, \{H, J, I\}, \{B, A, C, F\}, \{D\}, \{E\}\}$$

Argument

Trenger å vise

v og w er i den samme DFS treet av G^t. Da har vi $v \longrightarrow^* w \longrightarrow^* v$

Argument

Trenger å vise

La x være roten av en dfs tre av G^t og 2 noder v og w i det samme treet av G^t. Da har vi at

$$x \longrightarrow^* v \longrightarrow^* x \quad \text{and} \quad x \longrightarrow^* w \longrightarrow^* x$$

- x er en rot
  - $\Rightarrow$ $x \longrightarrow^* v$ in G^t
  - $\Rightarrow$ $v \longrightarrow^* x$ in G
- x har høyere post-order nummer enn v
  - $\Rightarrow$ x ble ferdig senere enn v i den første dybde-først traversering av G
  - $\Rightarrow$ v må være en etterkommer av x ellers vil v bli ferdig etter x.
  - $\Rightarrow$

$$x \longrightarrow^* v \in G$$

Oppsummering

Grafer: Oppsummering

- Implementasjon av grafer
- Topologisk sortering
- Korteste vei, uvektet graf
- Dijkstras alg. (korteste vei, vektet graf)
- Prim, Kruskal (minimale spenntrær)
- Dybde-først søk
- biconnectivity, SCC

Neste forelesning: 15. oktober

Kompleksitet og Algoritmeteori