

Graftraversering

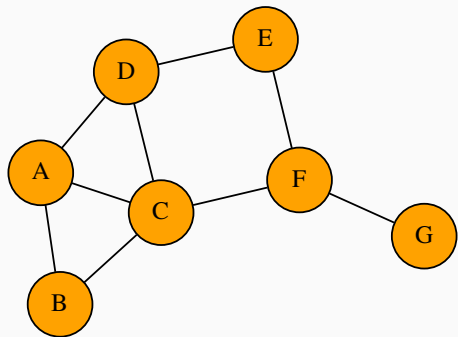
IN2010 – Algoritmer og Datastrukturer

Uke 38, 2020

Institutt for Informatikk

Graftraversering

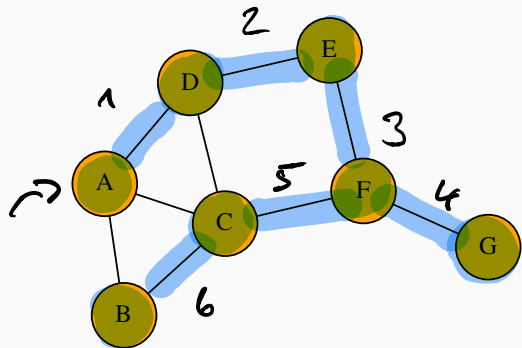
- måter å gå gjennom en graf
- hvilke noder kan nås fra en gitt startnode s ?
- hvilke komponenter har grafen?
- skal se på to algoritmer, tett knyttet til (1) stack og (2) kø.



Stack

Besøkt

A
D
E
F
C
B



Dybde-først søk (DFS)

Algorithm 1: Dybde-først søk

Input: En graf G og en startnode s

Output: Alle noder som kan nås fra s

```
1 Procedure DFS( $G, s$ )
2    $visited(s) = true$ 
3   for each edge  $(s, v)$  in  $G$  do
4     if  $visited(v) = false$  then
5       DFS( $G, v$ )
6   end
7 Procedure DFS-full( $G$ )
8   for each vertex  $v$  in  $G$  do
9     if  $visited(v) = false$  then
10      DFS( $G, v$ )
11  end
```

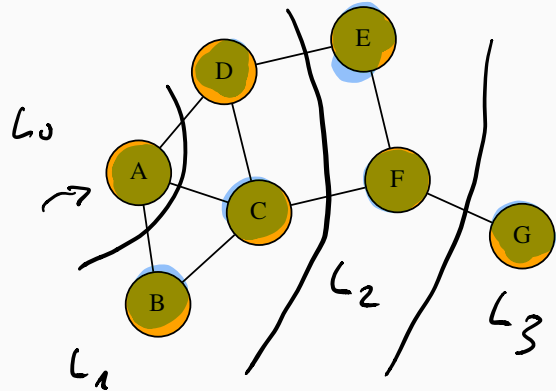
Analyse:

- DFS(G, s) besøker alle naboer av s , og rekursivt alle naboer av naboer, nøyaktig en gang Dette tilsvarer summen av $deg(v)$
- DFS-full sørger for at DFS blir kjørt på alle noder én gang.
- Dermed kjører algoritmen i $O(|V| + |E|)$ tid

Kø

Resultat

A
—
B
C
D
—
E
—
F
—
G



Bredde-først søk (BFS)

Algorithm 2: Bredde-først søk

Input: En graf G og en startnode s

Output: Alle noder som kan nås fra s

```
1 Procedure BFS( $G, s$ )
2   visited( $s$ ) = true
3    $L_0 = [s]$ 
4   while  $L_i$  not empty do
5      $L_{i+1} = []$ 
6     for each vertex  $v$  in  $L_i$  do
7       for each edge  $(v, w)$  do
8         if visited( $w$ ) = false then
9           visited( $w$ ) = true
10          add  $w$  to  $L_{i+1}$ 
11        end
12      end
13    end
14  end
```

Analyse:

- Samme argument som i dybde-først søk
- hver node blir besøkt en gang, som er i $O(|V|)$
- vi går gjennom nabolisten til hver node en gang, som er i $O(|E|)$
- tilsammen $O(|V| + |E|)$.