

## Øving 6 i algoritmer og datastrukturer, uvektede grafer

Det er to deloppgaver, bredde-først søk og topologisk sortering.

Gjør begge oppgavene. Grafene leses inn fra fil. Alle filene fungerer med bredde-først søk, ikke alle kan sorteres topologisk. Filformatet er det samme, det kan være lurt å implementere en felles grafklasse med metoder for å utføre de to oppgavene.

### Bredde-først søk

Implementer bredde-først søk (BFS). Programmet må kunne lese inn grafer fra fil, formatet for en graf uten vekter er slik:

#### Filformat

```
Nodeantall Kantantall
franode tilnode
franode tilnode
...
```

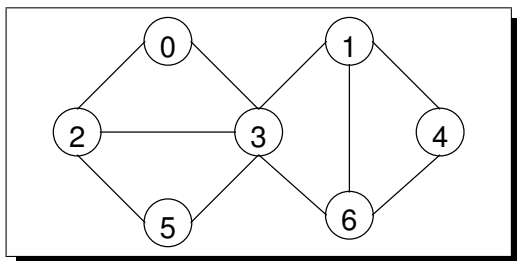
Formatet spesifiserer en rettet graf. For å få en urettet graf oppgis hver kant to ganger, en for hver retning.

#### Fil

```
7 20
0 2
0 3
1 3
1 4
2 0
2 5
2 3
3 0
3 5
3 2
3 1
3 6
4 1
4 6
5 2
5 3
6 3
6 1
6 4
1 6
```

#### Tegnet graf

Samme graf som i filen til høyre:



Etter søket må programmet legge frem resultatene, altså avstand til hver node fra startnode, og hver nodes forgjenger i BFS-treet. Eksempel hvis vi starter i node 5 i grafen over:

Node	Forgj	Dist
0	3	2
1	3	2
2	5	1
3	5	1
4	6	3
5		0
6	3	2

Under utprøving må programmet kunne kjøres med en hvilken som helst startnode, og en hvilken som helst av filene.

## Topologisk sortering

Implementer topologisk sortering. Programmet må skrive ut en brukbar rekkefølge for nodene. Én av flere gyldige rekkefølger for L7g5 er slik:

6 0 3 2 4 5 1

## Nedlastbare grafer

[http://www.iie.ntnu.no/fag/\\_alg/uv-graf/L7g1](http://www.iie.ntnu.no/fag/_alg/uv-graf/L7g1)

[http://www.iie.ntnu.no/fag/\\_alg/uv-graf/L7g2](http://www.iie.ntnu.no/fag/_alg/uv-graf/L7g2)

[http://www.iie.ntnu.no/fag/\\_alg/uv-graf/L7g3](http://www.iie.ntnu.no/fag/_alg/uv-graf/L7g3)

[http://www.iie.ntnu.no/fag/\\_alg/uv-graf/L7g5](http://www.iie.ntnu.no/fag/_alg/uv-graf/L7g5)

[http://www.iie.ntnu.no/fag/\\_alg/uv-graf/L7Skandinavia](http://www.iie.ntnu.no/fag/_alg/uv-graf/L7Skandinavia)

[http://www.iie.ntnu.no/fag/\\_alg/uv-graf/L7Skandinavia-navn.txt](http://www.iie.ntnu.no/fag/_alg/uv-graf/L7Skandinavia-navn.txt)

Fil	str	noder	kanter	
L7g1		7	20	Grafen på forrige side
L7g2	1 kB	50	100	Tilfeldige kanter
L7g3	39 kB	3 000	3 300	3000 noder i ring + 300 tilfeldige kanter
L7g5		7	11	Kan sorteres topologisk
L7Skandinavia	149 MB	4 426 216	10 046 924	Veikart
L7Skandinavia-navn		49 652	stedsnavn for veikartet	

## Noen praktiske tips

Det skal ikke ta «lang tid» å laste ned en fil på 149 MB. Men nettlesere liker ikke å vise frem en fil med ti millioner linjer. Å laste ned går fort, å vise frem et slikt monster går seint. *Så ikke klikk på den lenken!* Den er ikke ment for å vises frem i et brukergrensesnitt. De som bruker linux, kan bruke kommandoen «wget url» for å laste ned fila fort og greit. Andre kan høyreklikke og bruke en eller annen variant av «Lagre lenke som/Save Link As». Så blir filen lagret *uten* at nettleseren kaster bort sine beste år på å vise den frem. Dataingeniører skal ikke streve med å laste ned en fil på noen hundre MB. :-)

Filen med stedsnavn trenger ikke håndteres av programmet for å gjøre oppgaven. Den er med så interesserte vet hvilke stedsnavn som hører til nodene. Bredde-først søket kan finne den veien fra et sted til et annet som involverer færrest veikryss. Et par kjappe søk i fila viser at Drammen har node 65205 og Helsinki har node 3378527. Det er 990 veistykker (og dermed 899 kryss) mellom disse stedene. I motsatt retning er det 993 veistykker i mellom, enveiskjøringer og rundkjøringer skaper forskjeller.

Hvor mange kryss er det mellom Kalvskinnet og Moholt?