

Oving 5

Hvordan kjøre

```
make run del=1 // Hvis man vil kjøre del 1 (bfs)
make run del=2 // Hvis man vil kjøre del 2 (topologisk søk)
```

Del 1

Koden er skrevet med hensyn på en rettet graf da det står i oppgavebeskrivelsen "Formatet spesifiserer en rettet graf" og i tillegg tar boka hensyn i en rettet graf.

Siden koden tar hensyn til en rettet graf er det noen noder som ikke kan nås. Disse markeres med "not reachable" i utskriften. Verken startnoden eller noder som ikke kan nås har ingen forgjenger og "predecessor" feltet i utskriften blir derfor tom.

Bredde-først søk fungerer og gir rett resultat for de fem første grafene. (ø5g1, ø5g2, ø5g3, ø5g5 og ø5g7)

```
int main()
{
    FILE *fptr;
    // fptr = fopen("ø5g1.txt", "r");
    // fptr = fopen("ø5g2.txt", "r");
    // fptr = fopen("ø5g3.txt", "r");
    // fptr = fopen("ø5g5.txt", "r");
    // fptr = fopen("ø5g7.txt", "r");
}
```

For å teste ut grafene må man fjerne `//` foran filen man vil teste

Bredde-først søk kan kjøres med en hvilken som helst node som startnode.

```
int startNodeIdx = 0; // For bfs
```

Start noden sin indeks endrer man manuelt i `int main()`. Har vurdert å sette indeksen og graffilene i makefile som argument, men det kan bli litt komplisert å kjøre.

Utskrift graf5 med startnode lik 0

Node	Predecessor	Distance
0		0
1	2	2
2	0	1
3	0	1
4	2	2
5	2	2
6		not reachable

Del 2

Topologisk sortering gir rett resultat på ø5g5 og ø5g7. Det er flere mulige riktige sorteringer på begge.

Implementerte topologisk sortering med dfs fra eksempelet i boka.

```
int main(int argc, char **argv)
{
    FILE *fptr;
    // fptr = fopen("ø5g1.txt", "r");
    // fptr = fopen("ø5g2.txt", "r");
    // fptr = fopen("ø5g3.txt", "r");
    // fptr = fopen("ø5g5.txt", "r");
    // fptr = fopen("ø5g7.txt", "r");

    // fptr = fopen("ø5Skandinavia.txt", "r");
}
```

For å teste ut grafene må man fjerne `//` foran filen man vil teste

Utskrift graf5

0 6 3 2 5 4 1

Ulik gyldige rekkefølgen i eksempelet da jeg de mest sannsynlig ikke bruker dfs-basert implementasjon av topologisk sortering.