

# Øving 6

## Algoritmer og Datastrukturer

Henrik Halvorsen Kvamme  
9. oktober 2023

### Innhold

1	Introduksjon	2
2	Teori	2
3	Konklusjon	3

# 1 Introduksjon

Det er to deloppgaver. Første deloppgave ber deg utføre et bredde-først søk fra en startnode i en graf og printe ut alle nodene, deres avstand til startnode, og den forrige node for i søket til hver node.

Den andre deloppgaven ber deg topologisk sortere en graf. Det er ikke alltid mulig å topologisk sortere en graf, og i dette tilfellet printer jeg ut at det ikke er mulig å topologisk sortere grafen.

**Krav for godkjenning er:**

- Bredde-først søk fungerer og gir rett resultat for de fem første grafene. (ø6g1, ø6g2, ø6g3, ø6g5 og ø6g7) Det er ikke nødvendig å håndtere Skandinavia-filen nå, men den er eksempel på hvor store filer dere vil trenge å håndtere i senere øvinger.
- Bredde-først søk kan kjøres med en hvilken som helst node som startnode.
- Topologisk sortering gir rett resultat på ø6g5 og ø6g7. Det er flere mulige riktige sorteringer på begge.

# 2 Teori

Bredde-først søk begynner fra en startnode og sjekker iterativt alle naboer. Deretter sjekker den alle naboene til naboene, og så videre. Dette gjør at den finner alle noder som er på avstand 1 fra startnode, deretter alle noder som er på avstand 2 fra startnode, og så videre. Dette gjør at den finner den korteste veien fra startnode til alle andre noder i grafen. Bredde først tar ikke hensyn til lengde på kanter, og er derfor ikke optimalt for å finne korteste vei i en graf der ikke alle kanter er like lange. Da bruker man f.eks. Dijkstra's algoritme.

Topologisk sortering Handler om å sortere noder i en graf slik at alle kanter går fra venstre mot høyre. Det er ikke alltid mulig å topologisk sortere en graf, da vil algoritmen gi beskjed om dette. Det er kun mulig å topologisk sortere en graf som ikke har sykler (Directed Acyclic Graphs 'DAG'-grafer). En sykel er en vei i grafen som starter og slutter i samme node. Hvis en graf har en sykel, vil det ikke være mulig å topologisk sortere grafen.

Det er vanlig å bruke både bredde-først med Khans algoritme og DFS for topologisk sortering. Jeg velger å bruke **Khans algoritme**:

1. Finn alle noder som ikke har noen kanter inn til seg. Legg disse nodene i en kø. Antall kanter inn til en node er dens "indegree".

2. Ta ut en node fra køen og legg den i en liste. Fjern alle kanter ut fra denne noden. Hvis en node ikke har flere kanter inn til seg, legg den i køen.
3. Gjenta steg 2 til køen er tom.

Hvis listen ikke inneholder alle nodene i grafen, er det ikke mulig å topologisk sortere grafen. Ellers returner listen.

### **3 Konklusjon**

Jeg valgte å implementere en klasse for hver graf som tar inn filnavn for å initialiseres. Klassen har en metode for å løse for første deloppgave, breddeførst, og en for andre oppgave, topologisk sortering.

Kildekoden ligger vedlagt i main.cpp. Main funksjonen kjører flere tester for å vise at den utfører oppgaven.