

Oblig 2 - IN2010

Hvordan kjøre koden:

Pass på at **movies.tsv** og **actors.tsv** er i samme mappe som resten av Java filene. Kompiler alle Java-filene og kjør Oblig2 filen med: **Java oblig2**

Oppgave 1

Her leser vi inn alle skuespillere og filmer ved hjelp av et skanner-objekt. Vi lager objekter av alle skuespillerne og filmene og lager kant objekter, som kobler sammen skuespillere gjennom felles filmer de spiller i.

Kjøretiden på oppgave 1 er ca. **1.1 sekunder**

Oppgave 2

Her bruker Bredde-først-søk og stopper søket når vi kommer fram til riktig skuespiller, fordi det blir den korteste veien, uten at vi trenger å gå gjennom alle skuespillerne.

Kjøretiden på oppgave 2 er ca. **40 millisekunder**

Oppgave 3

Her bruker vi en Dijkstra metode for å gå gjennom og finne billigste vei fra en skuespiller til alle andre skuespillere. Denne metoden tar noe lengre tid enn oppgave 2, siden vi her må gå igjennom alle skuespillerne i hvert søk for å se om det finnes en billigere vei enn de vi allerede har funnet.

Kjøretiden på oppgave 3 er ca. **5 sekunder**

Oppgave 4

Her bruker vi et modifisert bfs som bruker en liste i stedet for en kø, siden rekkefølgen ikke har noe å si. Når vi har gått gjennom alle naboene til en tilfeldig startnode, så tar vi et nytt søk fra en ny tilfeldig node som ikke har blitt søkt enda, denne er da i en ny komponent osv.

Kjøretiden på oppgave 4 er ca. **1.8 sekunder**

Oppsummering

Kjøretiden på hele programmet bruker ca. **8 sekunder**

Merk at dette er tidene på en helt grei laptop med laderen plugget i. Vi har implementert en tidtaker i main-programmet som tar tiden på alle oppgavene og tiden programmet bruker totalt. Dette blir skrevet ut etter hver oppgave og skriver ut en liten oppsummering i slutten av programmet. Vi forventer at programmet kjører og løser alle oppgavene korrekt uten feil, men forventer at tiden vil variere noe fra kjøring til kjøring og pc til pc.