

# Oblig4 - Grafer

---

*Sist oppdatert av Atle Geitung 08.11.2018*



Øvingen skal gjennomføres i grupper på **2-4** studenter.

**Dere danner selv grupper i Canvas ved å registrere dere i en av de 40 forhåndsdefinerte gruppene «DAT108 Oblig4 gruppe x».** Legg helst til alle studentene i gruppen samtidig. (Gå inn på «Personer» | «DAT108 Oblig4 gruppe», og legg til dere selv i en tom gruppe)



Øvingen har veiledning på labbene E443/E403 fredagene t.o.m. 16. november. **Innleveringsfrist er om kvelden søndag 18. november.** Vi har som mål å rette innleveringene og godkjenne innen 2 uker.



**Innleveringen gjøres i Canvas.** Denne skal inneholde:

Et dokument (pdf) som

- lister opp hvem som er med i gruppen (for å unngå gratispassasjerer som melder seg inn i gruppe uten at det er avtalt).
- Svar på teorispørsmålene.

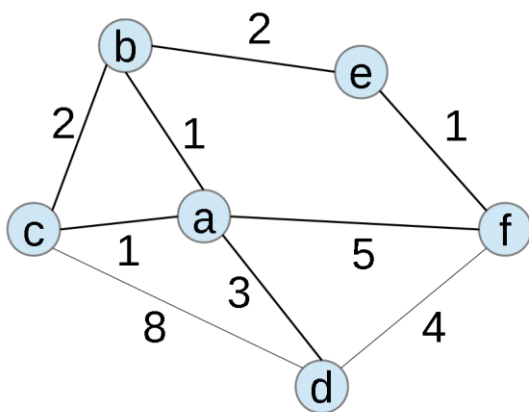
Et Eclipse-prosjekt (zip) med kildekode til koden.

## Oppgave 1

- a) Lag klassen Graf, du bestemmer selv hvilken implementasjon du vil bruke.
- b) Implementer graf-metodene fjernNode og fjernKant.
- c) Legg til og implementer metoden breddeFørst(Node n).
- d) Legg til og implementer metoden prim for å finne et minste spennetre og la den returnere en liste av kantene som definerer dette treet.

## Oppgave 2

- a) Forklar hvilke type problemer som problemklassene P, NP og NPC inneholder.
- b) Forklar hva som menes med en n-tilnærmingsalgoritme?
- c) «Reisende handelsmann»-problemet er et NP-komplett problem som er beslektet med «nodedekning»-problemet som består i å finne den korteste stien i en vektet graf og som besøker nodene i grafen nøyaktig en gang og som starter og slutter i samme node. En enkel tilnærmingsalgoritme for dette problemet er «nærmeste-nabo»-algoritmen. Pseudokode for algoritmen finner du under.
  - i. Hva er den korteste sykkelen i grafen under
  - ii. Hvilken sykel gir algoritmen som resultat dersom du starter i f?
  - iii. Hva sier dette om ordenen n til tilnærmingsalgoritmen.



Nærmeste nabo:

1. Velg en tilfeldig node som aktuell node.
2. Finn den korteste kanten som forbinder aktuell node med en ubesøkt node N.
3. Sett aktuell node til N.
4. Merk N som besøkt.
5. Dersom alle noder er besøkt, avslutt og returner en sekvens av noder.
6. Gå til steg 2.