

Øving 6: Trær

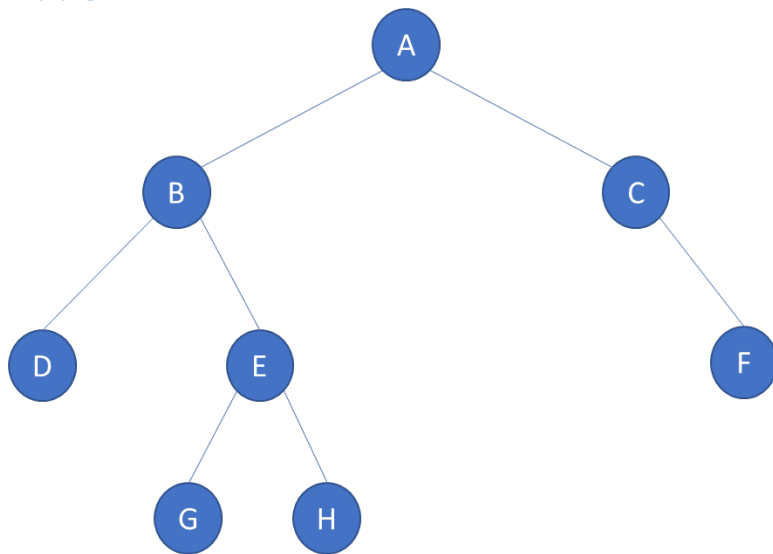
Læringsmål

Dere skal forstå hvordan trestrukturer virker og lære hvordan å bruke dem.

Godkjenning

Denne øvingen kan leveres i grupper på to studenter. Øvingen skal normalt sett godkjennes av studentassistent under øvingstimene.

Oppgave 1: Forståelse av trestrukturer



Gitt treet over

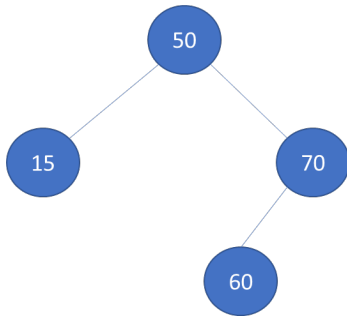
- Hvilken node er rotnoden?
- Hvilke noder er bladnoder?
- Fyll inn tabellen:

Node	Barn	Søsken	Dybde
A			
B			
C			
D			
E			
F			
G			
H			

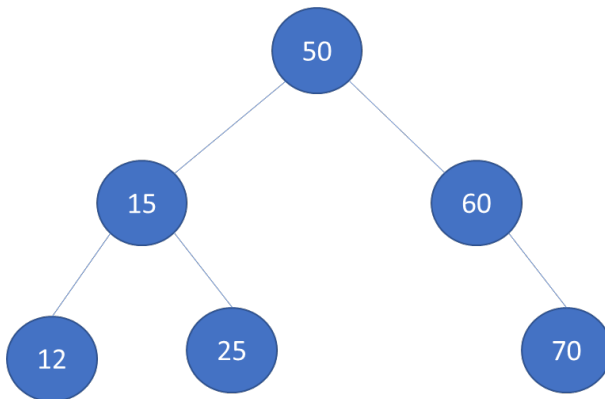
- Skriv nodene i treet i
 - Preorder rekkefølge

- b. Inorder rekkefølge
- c. Postorder rekkefølge

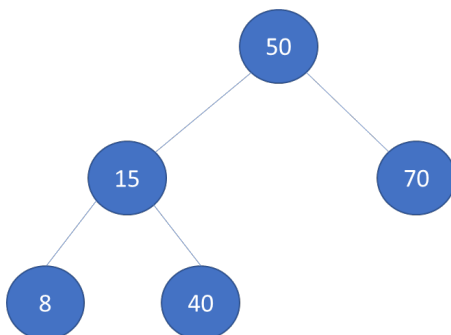
Oppgave 2: AVL trær



- a) Gitt AVL-treet over, som du kan anta er lovlig. Sett inn tallet 25.
 - a. Tegn AVL treet
 - b. Er AVL treet korrekt etter denne innsetningen? Hvis ikke, vis hvert steg i prosessen for å gjøre det om til et korrekt AVL tre



- b) Gitt AVL-treet over, som du kan anta er lovlig. Sett inn tallet 85.
 - a. Tegn AVL treet
 - b. Er AVL treet korrekt etter denne innsetningen? Hvis ikke, vis hvert steg i prosessen for å gjøre det om til et korrekt AVL tre



- c) Gitt AVL-treet over, som du kan anta er lovlig. Sett inn tallet 35.
 - a. Tegn AVL treet
 - b. Er AVL treet korrekt etter denne innsetningen? Hvis ikke, vis hvert steg i prosessen for å gjøre det om til et korrekt AVL tre

Oppgave 3: Bruk av trestrukturer

- a) Med utgangspunkt i binærtreimplementasjonen fra timen (binaertre.py), skriv en rekursiv metode som beregner antall bladnoder i treet.
- b) Lag en preorder iterator for basis binærtreet fra forelesningene (binaertre.py). Du kan basere deg på postorder iteratoren fra timene, men trenger ikke det. Preorder iteratoren kan gjøres betydelig enklere enn postorderiteratoren er.
- c) Bruk avl-treet fra timene til å Implementere finn-k-ende-verdi algoritmen som ble presentert i videoen «7 trestrukturer 11 AVL trær k minste verdi algoritme», inkludert å legge til egenskapen «størrelse» til hver node i treet.
 - a. Obligatorisk: Vedlikehold «størrelse» egenskapen under innsettinger, inkludert rotasjoner
 - b. Obligatorisk: Lag metoden finn_k_ende(k) som finner k-ende minste verdi.
 - c. **Frivillig:** Vedlikehold «størrelse» egenskapen under sletting
- d) Gitt fila vindmaalinger_redusert_mer.txt samt utdelt kode for å lese inn denne fila. Bruk en av AVL-tre implementasjonene fra timene for å lage et map over vindmålinger for å gjøre følgende spørringer, samt implementer spørringene:
 - a. Finn hvor sterk vinden var midt på dagen (klokka 12:00) den 4. september 2010. Det holder å returnere nærmeste vindmåling etter det oppgitte tidspunktet.
 - b. Finn hvor sterk vinden var på sitt kraftigste den 7. desember 2011. Du kan gjøre dette ved å spørre om alle vindmålinger denne datoen (fra 7. desember 2011 klokka 00:00 til 8. desember 2011 klokka 0:00), og så finne maksimum gjennom å gå gjennom denne lista.
 - c. **Frivillig:** Utvid koden slik at bruker kan oppgi tidspunktene for oppgave a og b. Merk at fila bare inneholder målinger fra årene 2009 til 2012. Du kan for eksempel bruke datoen 2011-12-25 (tidspunktet Dagmar herjet på nordvestlandet og Trøndelag)
- e) **Frivillig:** Lag en levelorder iterator for basis binærtreet fra forelesningene
- f) **Frivillig:** Filsystemet på en typisk datamaskin er formet som et tre hvor du har en eller flere rot-mapper («C:\» på en windows-maskin, «/» på en mac eller unix maskin), hvor hver mappe kan inneholde en kombinasjon av vanlige filer (bladnoder) og andre mapper (interne noder). Skriv et Python program som beregner størrelsen til alle filene som ligger under en oppgitt mappe, inkludert filer som ligger i undermappene. Merk: For å skaffe informasjon om data slik som størrelsen til filer, må dere importere pathlib, som er et mer avansert bibliotek for å behandle filer enn det som ligger inne standard.