

# Øving 8: Hauger

## Læringsmål

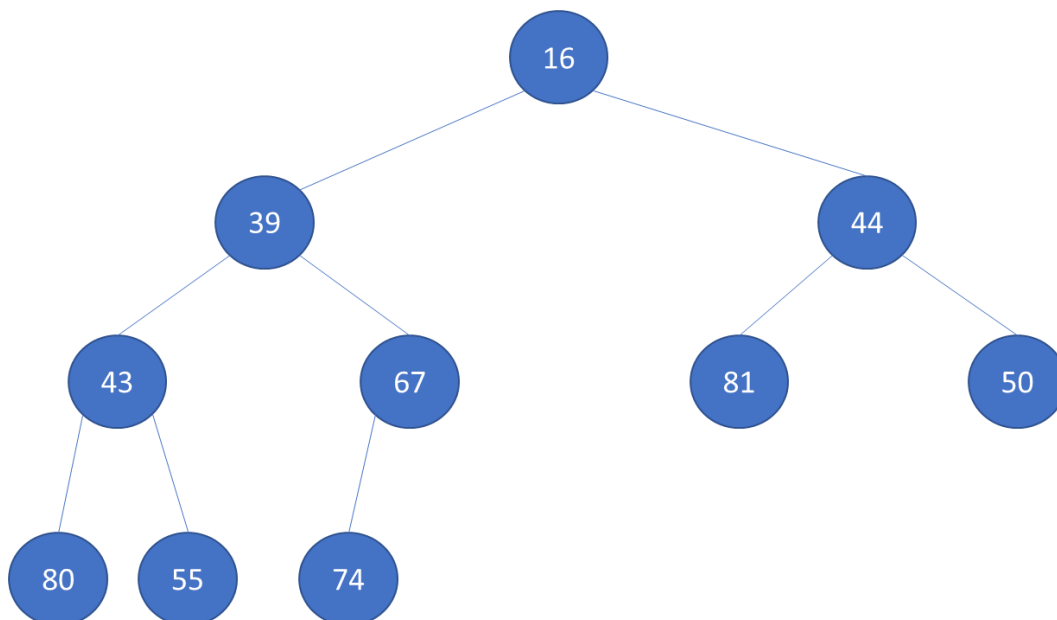
Dere skal få forståelse for hauger gjennom å se på noen eksempelhauger og se på bruken av hauger.

## Godkjenning

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

## Oppgave 1: Hvordan hauger virker

- Sett at du har haugen oppgitt under, som er en minimumhaug. Sett inn elementet «25» i haugen. Vis hvert steg i algoritmen som finner korrekt plassering for det nye elementet.
- Gitt samme haug (uten innsettingen fra oppgave a). Brukeren kjører `delete_min` på haugen. Vis hvert steg i operasjonen for å få tilbake en korrekt haug.



## Oppgave 2: Implementer Maksimumhaug

Lagt ved oppgaven er en implementasjon av en minimumshaug. Skriv om denne haug-implementasjonen til å være en maksimumhaug i stedet for en minimumhaug. Det vil si at i stedet for å finne og eventuelt fjerne minste verdi så skal den finne og eventuelt fjerne største verdi.

## Oppgave 3: Diskret hendelsessimulering

Skriv en diskret hendelsessimulator (eller skriv om eksemplet som er lagt ved) som simulerer et fergeleie. Målet er å finne ut hvor stor ferge man trenger og hvor stort område for bilene som venter på ferga. Ferga ankommer en gang i halvtimen. Man kan bruke ei lita ferge som tar 40 personbiler, ei mellomstor som tar 80 eller ei stor som tar 200. Om sommeren i den mest trafikkerte tida kommer det i gjennomsnitt to biler i minuttet, men bilene ankommer tilfeldig. Dessuten er ikke alle bilene like store. Anta at 75% av bilene er personbiler. Anta at 10% er bobiler som tar plass som 2 personbiler. Anta at 5% er busser som tar plass som 4 personbiler. Anta at de siste 10% er lastebiler som tar plass fra 3 til 8 personbiler.

Spørsmålet er:

1: Hvor stor ferge trenger man

2: Hvor stor plass må man sette av ved kaia for at bilene skal stå å vente.

Skriv en diskret hendelsessimulator som svarer på disse spørsmålene. Lengden på området bilene venter på kan telles opp hver gang ferga ankommer. Merk at ferga bare tar inn hele biler, så hvis det er igjen 2 plasser og neste kjøretøy er en lastebil med lengde 6, så blir den stående igjen på kaia til neste anløp.

**Frivillig:** Kjøretøyene har ulik prioritet. Lastebiler og busser har høyest prioritet etterfulgt av personbiler etterfulgt av bobiler.

**Frivillig:** Skriv om eksemplet fra timene slik at det er separate køer for hver skranke. Når det ankommer en kunde så stiller vedkommende seg i den korteste køen. Den som betjener en skranke, betjener bare dem som står i riktig kø. Sjekk om dette har noe å si for gjennomstrømningen i systemet.

## Oppgave 4, frivillig: HeapSort

I tillegg til å brukes som prioritetskøer kan hauger også brukes til sortering. Denne algoritmen kalles ofte heapsort. Bruk implementasjonen av en haug fra timene for å implementere heapsort. Finn ut hva kjøretida for implementasjonen din er i O-notasjon. Finn ut om heapsort er in-place og/eller stabil.