# Oblig 1
# IN2010 Høst 2020
# siljejka

## Oppgave 1

**b)** Push_front of push_back legger begge til elementet x først eller sist i listen, dette krever ikke noen iterasjoner or har derfor kompleksitet O(1).

Push_middle er O(n) for første input, men er etter det O(1) da den legger til elementet x før eller etter tidligere referase til midpunktet.

Metoden get itererer gjennom listen til den finner indeksen - og har derfor kompleksiteten O(n), der n er halvparten av lengden på listen.

**c)** Hvis det er en øvre begrensing på N så vet man at værstetilfellet har en øvre grense på antall operasjoner, og man kan se at kjøretiden er O(1).

## Oppgave 2

Å hente et element fra en lenked liste har kjøretiden O(n), og binærsøk har kjøretiden O(log n), dermed er værstetilfellet for binærsøk på en lenked liste O(n log n). Det ville vært bedre å bruke en annen algoritme for å finne ut om et element er i en lenked liste, siden å iterere gjennom hele listen har kjøretid O(n), og det er unødvendig å iterere gjennom listen flere ganger.

Hvis man bruker binærsøk på en datastruktur som har en mindre kompleksitet på henting av elementer så ville kjøretiden vært redusert. En sortert array liste er da bedre siden den har O(1) i kjøretid, og da blir værste tilfelle for alogritmen med binærsøk O(log n).