Oppgave 1a:

Graphical user interface, text, application

Description automatically generated

Oppgave 1c:

push\_front()

🡪 O(n) worst case, fordi alle elementene må i worst case kopieres til et nytt array

push\_middle()

🡪 O(n) worst case, fordi alle elementene må i worst case kopieres til et nytt array

push\_back()

🡪 O(n) worst case, fordi alle elementene må I worst case kopieres til et nytt array

get()

🡪 O(1) alltid lineært, fordi arraylist støtter direkte oppslag på indeks

Oppgave 1d:

Hvis vi vet at N er begrenset, betyr det at algoritmen kjøres i konstant tid.

Oppgave 2:

I et verste- tilfelle- estimat, når x ikke er i lista, vil algoritmen gi O(log(n)) fra binærsøket, og O(n) fra get- operasjonen til lenkede lister. Altså O(n\*log(n)). Hvis get- operasjonen hadde vært logaritmisk/ konstant, kunne algoritmen spart tid.

Oppgave 3a:

Text

Description automatically generated

Oppgave 4a:

Text

Description automatically generatedText

Description automatically generated

Oppgave 4b:

Text

Description automatically generated

Text

Description automatically generated