

IN2010 – Algoritmer og datastrukturer

Trym Bø og Herman Nordaunet

Obligatorisk oppgave 1, 18.09.2020



Contents

Oppgave 1	3
b	3
c	3
Oppgave 2	3

Oppgave 1

b

For operasjon `push_back` vil vi ha en verste-tilfelle kjøretid på konstant tid der vi har $O(1)$. Det samme gjelder for `push_front`. Dette er fordi uansett hvor mange noder som er i køen vil dette ikke endre antall operasjoner i `push_back` og `push_front`. Når det kommer til `push_middle` vil det være en verste-tilfelle kjøretid på $O(n)$ altså lineær tid. Dette selvom vi bare sjekker for halvparten av alle nodene vil vi abstrahere bort $n/2$ til n . Dette kommer av at vi i verste fall må sjekke gjennom alle nodene i køen. Det samme gjelder `get` for køen ettersom vi halverer antall noder vi løper igjennom, men vi abstraherer også her til $O(n)$ istedenfor $O(n/2)$.

c

Når vi skal se på verste-tilfelle kjøretid og vet at N er begrenset vil ikke dette ha så mye å si for vår kompleksitet i O -notasjonen. Dette ville for eksempel hatt noe å si hvis vi i vår O -notasjon hadde hatt $n^2 + n$ og brukt det største leddet for O -notasjonen n^2 . Da ville vi hatt $O(n^2)$. Ved begrenset N vil her $+n$ leddet ha mye å si for kjøretid. Da ville kompleksiteten av O -notasjonen blitt påvirket. Dette skjer ikke for oss ettersom vi ikke har noen operasjoner der vi velger det største leddet for O -notasjon.

Oppgave 2

Vår `get`-funksjon har worst case på $O(n)$ Binærsøk har tradisjonelt worst case på $O(\log(n))$

Siden `get`-funksjonen kalles i hvert steg av binærsøkalgoritmen, må vi gangen disse worst-case tidene sammen for å finne worst case for binærsøk ved hjelp av vår lenket liste.

Vi får da worst case $O(n \log(n))$