

Algoritmer og datastrukturer – O4g

Markus Dalbakk, Brage Tiller Naustan & Emil Leonard Aastrøm

- Vi har valgt å gjøre oppgave 1 og 3
- I den første oppgaven har vi i tillegg et krav om å regne på hva slags kompleksitet algoritmen vår har fått
 - Ettersom vi traverserer gjennom hele listen opp til flere ganger har vi kommet fram til at kompleksiteten i vårt tilfelle har en øvre grense på $O(n*m)$, hvor n er lengden på listen og m er det konstante intervallet mellom hver soldat som faller. I tilfellet hvor intervallet er lik lengda på listen vil man ende opp med å traversere $n*m$ ganger ettersom man kun fjerner en node for hver gang man traverserer den sirkulære listen.
 - Teoretisk sett om man setter intervallet til 1 ender man opp med å fjerne alle på første runde, og man får en nedre grense på $\Omega(n)$. I vårt tilfelle ender man opp med å støte på et problem siden vi bruker den forrige noden for å hoppe over nåværende.
- Vi løste oppgave 3 ved å traversere treet to ganger, en gang for å beregne summen av uttrykkstreet, og en gang for å gjøre uttrykkstreet til en String.
 - b) For å traversere uttrykkstreet generelt ville vi bruke postordentraversering. På denne måten kan vi identifisere og gjennomføre utregningen av de ulike operasjonene i rett rekkefølge.
 - c) Når man skriver opp uttrykkstreet som et vanlig aritmetisk uttrykk som for eksempel: $(3*(2+4))/(7-2(2*2))$, så er det skrevet opp som med en inordens traverseringsmåte. Dette fordi vi deler opp uttrykket i venstre/høyre 'barn' og håndterer regneoperasjonen mellom disse to.

- Annet eksempel: $3+5$.
 - 3 er venstre barn, pluss blir foreldrenoden, 5 er høyre barn