Algdat Excercise 2

Tomas Beranek and Callum Gran

August 2022

Første del av øvingen ble gjennomført ved å følge den matematiske likningen som ble gitt i oppgavebeskrivelsen til 2.1-1. Dette er en lineær rekursiv funksjon. Derfor er tidsforbruket gitt ved:

$$T(n) = \begin{cases} 1 & \text{når n} = 1 \\ T(n-1) + 1 & \text{når n} > 1 \end{cases}$$

Ut ifra denne likningen kan vi se at dette er en lineær rekusjon der tidskompleksiteten blir

$$\theta(n) \tag{1}$$

Andre delen av oppgaven ble løst ved bruk av en ny matematisk likning gitt av oppgavesettet 2.2-3. For å finne kompleksiteten av denne bruker vi master teorem. Den sier:

$$T(n) = \begin{cases} 1 & \text{når n} = 0 \\ aT(n/b) + f(n) & \text{når n} > 1 \end{cases}$$

Denne funksjonen ender opp med to rekursive kall, men kun et kalles av gangen. Dette betyr at det kun telles som et rekursivt kall. Eksponenten blir også delt på 2 hvor hver gang, og selv om det er blir fratrekk i det ene kallet, så ganges den før uansett. Master teorem har 3 tilfeller, hvor denne likningen passer tilfelle 2.

$$f(n) = \text{c } *n^0 \text{ for funksjonen.}$$

$$\theta(n^0) = \theta(n^{\log_b a})$$
$$\theta(n^0) = \theta(n^{\log_2 1})$$
$$\theta(n^0) = \theta(n^0)$$

Derfor er er tidskompleksiteten til funksjonen:

$$T(n) = \theta(n^{\log_b a} * \log n)$$
$$T(n) = \theta(n^0 * \log n)$$
$$T(n) = \theta(\log n)$$

Utifra utregningen vi har fått kan vi se at den andre algoritmen, eller den andre likningen, tar mye kortere tid enn den første (siden kjøretiden er i log(n) istedenfor n). Utifra måligene under kan vi se at utregningen stemmer, og dermed også analysen av tidskompleksiteten.

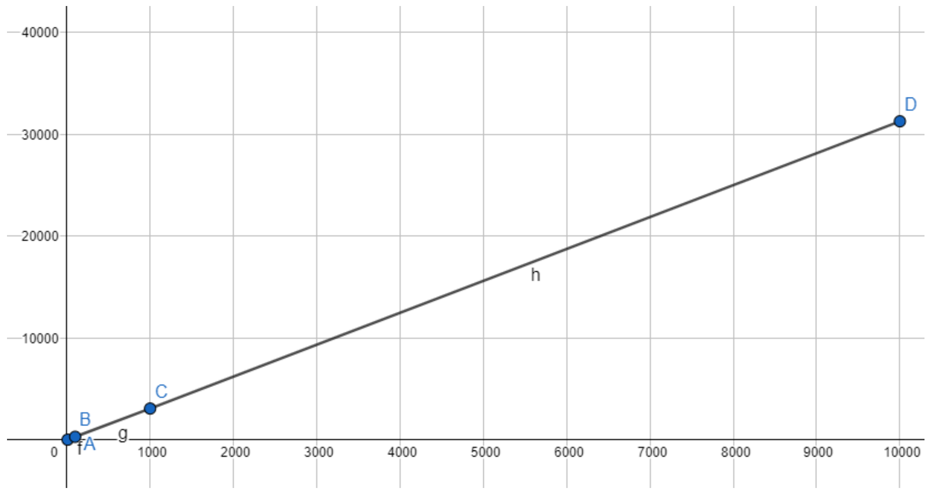

Figure 1: bilde over $\theta(n)$

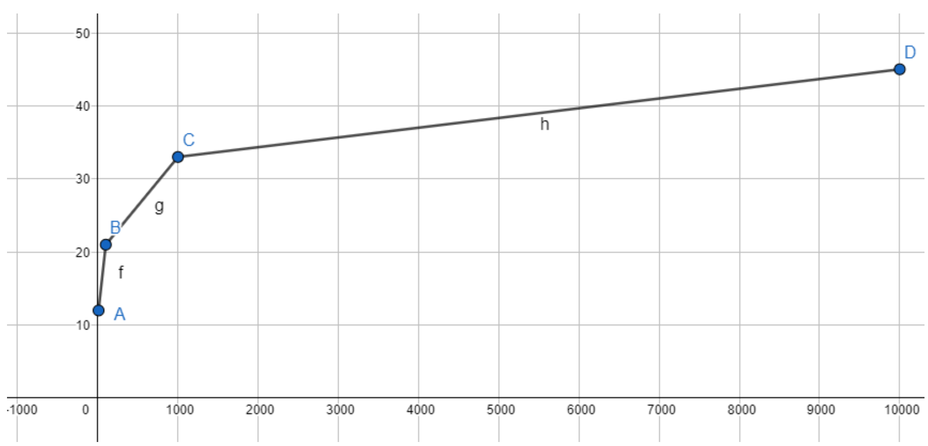

Figure 2: bilde over $\theta(log(n))$

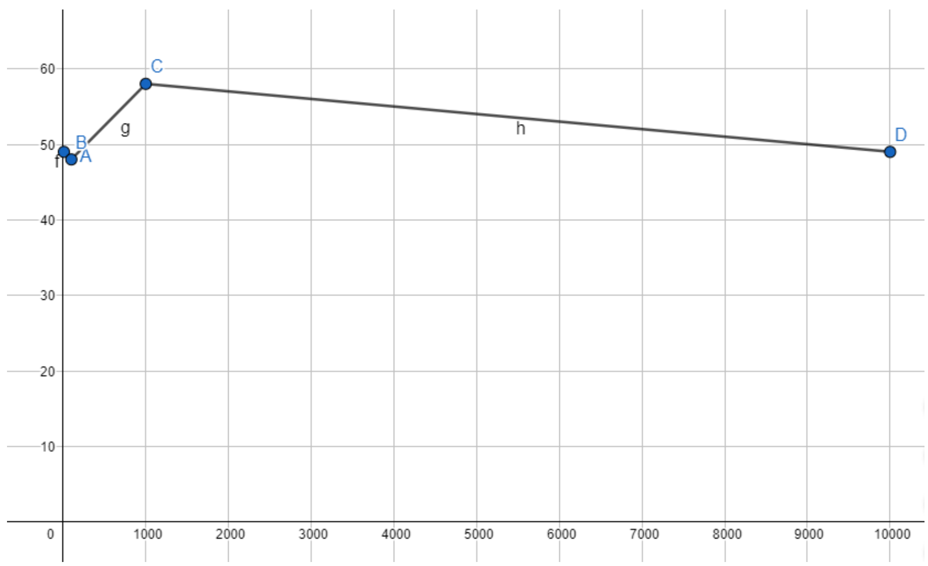

Figure 3: bilde over $\theta(1)$