

Analyse av Algoritmers Tidskompleksitet

Medlemer: Thobias Høivik, Gunnar Salbu, Birk Tangen, Glenn Boine

01/29/2025

1 Oppgaver

1.1 Oppgave 3a

Bestem størrelsesorden uttrykt i O-notasjon for følgende vekstfunksjoner:

1. $4n^2 + 50n - 10 \quad O(n^2)$
2. $10n + 4 \log_2 n + 30 \quad O(n)$
3. $13n^3 + 22n^2 + 50n + 20 \quad O(n^3)$
4. $35 + 13 \log_2 n \quad O(\log n)$

1.2 Oppgave 3b

Gitt algoritmen:

```
sum = 0;
for (int i = n; i > 1; i = i/2) {
    sum = sum + i;
}
```

Antall tilordninger er $O(\log n)$, siden i halveres i hver iterasjon.

1.3 Oppgave 3c

Gitt algoritmen:

```
sum = 0;
for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= n; j = j * 2) {
```

```

        sum += i * j;
    }
}

```

Indre løkke kjører $O(\log n)$ ganger for hver av de n ytre iterasjonene, gir totalt $O(n \log n)$.

1.4 Oppgave 3d

Arealet til en sirkel er $A = \pi r^2$, som er $O(r^2)$. Omkretsen er $C = 2\pi r$, som er $O(r)$.

1.5 Oppgave 3e

Gitt metoden:

```

boolean harDuplikat(int tabell[], int n) {
    for (int indeks = 0; indeks <= n - 2; indeks++) {
        for (int igjen = indeks + 1; igjen <= n - 1; igjen++) {
            if (tabell[indeks] == tabell[igjen]){
                return true;
            }
        }
    }
    return false;
}

```

I verste fall må vi sammenligne alle par, som gir $O(n^2)$.

1.6 Oppgave 3f

1. $t_1(n) = 8n + 4n^3 \quad O(n^3)$
2. $t_2(n) = 10 \log_2 n + 20 \quad O(\log n)$
3. $t_3(n) = 20n + 2n \log_2 n + 11 \quad O(n \log n)$
4. $t_4(n) = 4 \log_2 n + 2n \quad O(n)$

Rangering fra best til verst: $O(\log n) < O(n) < O(n \log n) < O(n^3)$.

1.7 Oppgave 3g

Metoden:

```
public static void tid(long n) {  
    long k = 0;  
    for (long i = 1; i <= n; i++) {  
        k = k + 5;  
    }  
}
```

Metoden har n operasjoner, så $T(n) = O(n)$. Målinger for $n = 10^7, 10^8, 10^9$ bør reflektere lineær vekst.