

Oblig 1 – Joachim Leiros – DAT102

Oppgave 1

Se kode.

Oppgave 2

a)

$$1 - 4n^2 + 50n - 10 = O(n^2)$$

$$2 - 10n + 4\log_2 n + 30 = O(n)$$

$$3 - 13n^3 - 22n^2 + 50n + 20 = O(n^3)$$

$$4 - 35 + 13\log_2 n = O(\log_2 n)$$

b)

Ettersom algoritmen kun har en tilordning da kun addisjon blir brukt så har den 1 tilordning. Algoritmen blir da uttrykt som $O(n^2)$.

c)

	A	B	C
Tilordning	N	n^2	1
Addisjon	$1(\text{sum}+i)$	$1(\text{sum}+1)$	1
Multiplikasjon	0	0	1
Divisjon	0	0	1
Totalt:	n	N^2	3

d)

$T(n)$	$T(10^6)/10/6$
$\log_2 n$	$6 \cdot 10^{-6}$ sekunder
N	1 sekund
$N \log_2 n$	6 sekunder
N^2	11.57 dager
N^3	31709.79198 år

e)

f)

$$1 - 8n + 4n^3 = O(n^3)$$

$$2 - 10\log_2 n + 20 = O(\log_2 n)$$

$$3 - 20n + 2n\log_2 n + 11 = O(n)$$

$$4 - 4\log_2 n + 2n = O(\log_2 n)$$

Den mest effektive algoritmen er 3, $O(n)$ og den minst effektive er 1, $O(n^3)$.

g)

```
package no.hvl.dat102;
public class Main {
    public static long tid(long n) {
        long starttid = System.currentTimeMillis();
        long k = 0;
        for (long i = 1; i <= n; i++) {
            k = k + 5;
        }
        return System.currentTimeMillis() - starttid;
    }
    public static void main(String[] args) {
        long[] snitt = new long[3];
        int antallMaalinger = 20;
        for (int i = 0; i < antallMaalinger; i++) {
            snitt[0] += tid( n: 1000000000L);
            snitt[1] += tid( n: 10000000000L);
            snitt[2] += tid( n: 100000000000L);
        }
        for (int i = 0; i < snitt.length; i++) {
            System.out.println("'" + i + ": " + (snitt[i]/antallMaalinger + "ms"));
        }
    }
}
```

Målinger:

```
0: 43ms
1: 390ms
2: 3824ms
```

Oppgave 3

1 – $O(n)$

2 -

Oppgave 4

a)

Test Results	74 ms	Test Results	55 ms
StabelADTTest	74 ms	StabelADTTest	55 ms
pushPopErTom()	39 ms	pushAndPopWithDuplicate	26 ms
erIkkeTom()	7 ms	pushPopIsEmpty()	1 ms
nyStabelErTom()	3 ms	newStackIsEmpty()	8 ms
pushOgPopMedDuplikater()	12 ms	pushPopPushPushPopPeek()	7 ms
pushOgPop()	3 ms	isNotEmpty()	3 ms
stor()	1 ms	pushAndPop()	3 ms
pushPopPushPushPopPeek()	1 ms	utviderKoeSeg()	1 ms
popFromEmptyIsUnderflow	8 ms	popFromEmptyIsUnderflow	6 ms

b)

- 1- Tester om en stabel med null elementer er tom. Kaster exception om den ikke er det.
- 2- Kaster exception om det blir prøvd å poppe på en tom stabel.

c)

<default package>	122 ms
KjedetKoeTest	81 ms
InnkoeOgUtkoeMedG	10 ms
testToString	
InnOgUtkoeForst	
erTom	
InnkoeUtkoeerTom	
antallTest	
innogutKoe	
UtkoeErTom	71 ms
SirkulaerKoeTest	15 ms
InnkoeOgUtkoeMedG	1 ms
testToString	1 ms
InnOgUtkoeForst	3 ms
erTom	1 ms
InnkoeUtkoeerTom	
antallTest	1 ms
innogutKoe	
UtkoeErTom	8 ms
TabellKoeTest	26 ms
InnkoeOgUtkoeMedG	1 ms
testToString	10 ms
InnOgUtkoeForst	9 ms
erTom	1 ms
InnkoeUtkoeerTom	2 ms
antallTest	1 ms
innogutKoe	1 ms
UtkoeErTom	1 ms