

# DAT102 Oblig1

Gruppe: 65

Deltagere: Kjartan Gundersen

```
Velkommen til Filmarkiv!

Valgmeny:
1 - Legg til film
2 - Slett film
3 - Søk etter filmtittel
4 - Søk etter filmprodusent
5 - Skriv ut statistikk
Valg: 5

Antall filmer totalt: 10
Antall actionfilmer: 6
Antall dramafilmer: 4
Antall Sci-Fi filmer: 0

Valgmeny:
1 - Legg til film
2 - Slett film
3 - Søk etter filmtittel
4 - Søk etter filmprodusent
5 - Skriv ut statistikk
Valg: 2
Hvilket filnummer vil du slette?: 2
Filmen ble slettet!

Valgmeny:
1 - Legg til film
2 - Slett film
3 - Søk etter filmtittel
4 - Søk etter filmprodusent
5 - Skriv ut statistikk
Valg: 3

Søk etter filmtittel: Iron

FilmNr: 9
Produsent: MCU
Tittel: Iron Man
Utgivelsesår: 2008
Sjanger: Action
Filmsekskap: Marvel

Valgmeny:
1 - Legg til film
2 - Slett film
3 - Søk etter filmtittel
4 - Søk etter filmprodusent
5 - Skriv ut statistikk
Valg:
```

```
12 FilmarkivADT testFilmArkiv = new Filmarkiv2(new LinearNode<Film>(film1));
11
10 @Test
9 void testConstructor() {
8     assertNotNull(testFilmArkiv);
7     testFilmArkiv.leggTilFilm(film2);
6     assertEquals(2, testFilmArkiv.antall());
5 }
4
3
2
1 void testFinnFilm() {
30     assertEquals(null, testFilmArkiv.finnFilm(0));
1     testFilmArkiv.leggTilFilm(film2);
2     assertEquals(film1, testFilmArkiv.finnFilm(1));
3 }
4
5 @Test
6 void testAntall() {
7     assertEquals(1, testFilmArkiv.antall());
8 }
9
10 @Test
11 void testLeggTilFilm() {
12     assertEquals(1, testFilmArkiv.antall());
13     testFilmArkiv.leggTilFilm(film2);
14     assertEquals(2, testFilmArkiv.antall());
15 }
16
17 @Test
18 void testSlettFilm() {
19     testFilmArkiv.leggTilFilm(film2);
20     testFilmArkiv.leggTilFilm(film3);
NORMAL ▶ P main > -1 ~/Projects/HVL/DAT102/Arbeidskrav_1/src/no/hvl/data102/filmarkiv/test/Filmarkiv2Test.java utf-8 < Δ < 4 java 37% 30:23
7 ✓testSoekTittel
6 ✓testLeggTilFilm
5 ✓testSoekProdusent
4 ✓testFinnFilm
3 ✓testSlettFilm
2 ✓testAntall
1 ✓testAntallSjanger
9 ✓testConstructor
```

[dap-repl] [+]  
Tests finished. Results printed to dap-repl. All 8 succeeded

9:1

## Oppgave 3

a)

- i.  $O(n^2)$
- ii.  $O(n)$
- iii.  $O(n^3)$
- iv.  $O(\log n)$

b)

I algoritmen ser vi at det er en loop som går fra “n” størrelse og ned mot 1. For hver repetisjon av løkken divideres indeksen med 2, noe som betyr at utviklingen ikke er eksponentiell, lineær eller konstant. Repetisjonen av løkken har en mengde på logaritmen på n og kan noteres ved  $O(\log n)$ .

c)

Denne algoritmen består av to løkker, ytre og indre. Den ytre løkken har en lineær økning og vil da kjøre n ganger. Den indre løkken er logaritmisk og kan noteres med  $O(\log n)$  som forrige oppgave. Men siden den indre løkken blir repetert av den ytre løkken vil notasjon på hele algoritmen bli  $O(n \log n)$ .

d)

Dersom sirkelen vokser så er det r (radius) som endres. Det vil si at  $2\pi$  er konstant, det som endres vil da være  $r^2$  og r. Den tyngste av disse vil være den som noteres med i O-notasjonen og det vil dermed bli  $O(n^2)$ .

e)

I denne algoritmen så vil i verste tilfelle vi finne de to siste i tabellen som duplikater. Det vil bety at ytre løkken vil måtte kjøre  $n-2$  ganger, men for hver gang den ytre løkken repeteres så vil den indre løkken kjøre  $n-2$  ganger. Det vil si hele ytre og indre løkken repeteres  $(n - 2)^2$  ganger. Effektiviteten vil da være  $O(n^2)$ .

f)

- i.  $O(n^3)$
- ii.  $O(\log n)$
- iii.  $O(n \log n)$
- iv.  $O(n)$

I rangert rekkefølge hvor 1 er raskest og 4 er tregeest kan funksjonene rangeres slik:

- 1. ii
- 2. iv
- 3. iii
- 4. i