

---

# Sorteringsalgoritmer og O-notation

---

Matematik A og Informatik C

Vejledere: Jens Christian Larsen og Kristian Kjeldgaard Hoppe

Balder Westergaard Holst

24. marts 2022

# Indhold

<b>1</b>	<b>Indledning</b>	<b>2</b>
<b>2</b>	<b>Resume</b>	<b>3</b>
<b>3</b>	<b>Store-O-Notation</b>	<b>4</b>
<b>4</b>	<b>De to Algoritmer</b>	<b>5</b>
4.1	Insertionsort . . . . .	5
4.1.1	Analyse af Insertionsort . . . . .	5
4.2	Mergesort . . . . .	5
4.3	Sammenligning af Algoritmerne . . . . .	6
4.4	Den Optimerede Mergesort . . . . .	6
<b>5</b>	<b>Den Hurtigste Køretid</b>	<b>7</b>
<b>6</b>	<b>Konklusion</b>	<b>8</b>

# 1 | Indledning

Dette er min indledning pt.

## 2 | Resume

Dette er hvad jeg har skrevet og fundet ud af.

## 3 | Store-O-Notation

$$O(f(n)) = \{g(n) : \exists c > 0 : \exists n_0\}$$

## 4 | De to Algoritmer

### 4.1 Insertionsort

Insertionsort implementeret i python:

---

```
def insertionsort(l):
    for i in range(1, len(l)):
        element = l[i]

        if element < l[0]:
            for j in range(i, 0, -1):
                l[j] = l[j-1]
            l[0] = element
        else:
            j = i
            while(l[j-1] > element):
                l[j] = l[j-1]
                j -= 1
            l[j] = element
    return(l)
```

---

#### 4.1.1 Analyse af Insertionsort

Funktionen for køretiden af denne algoritme er en del af mængden  $O(n^2)$ .

### 4.2 Mergesort

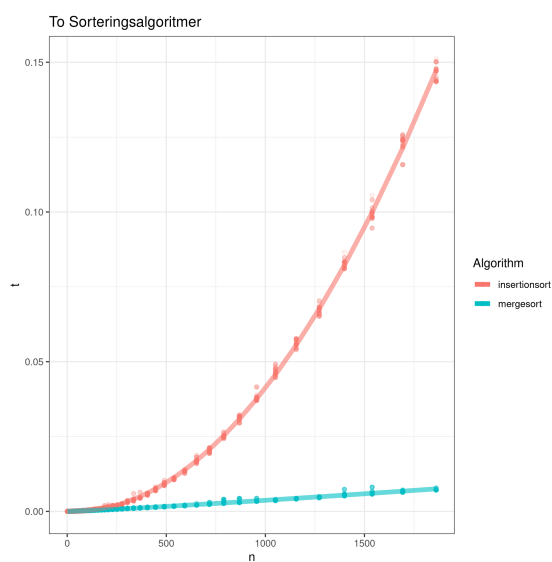
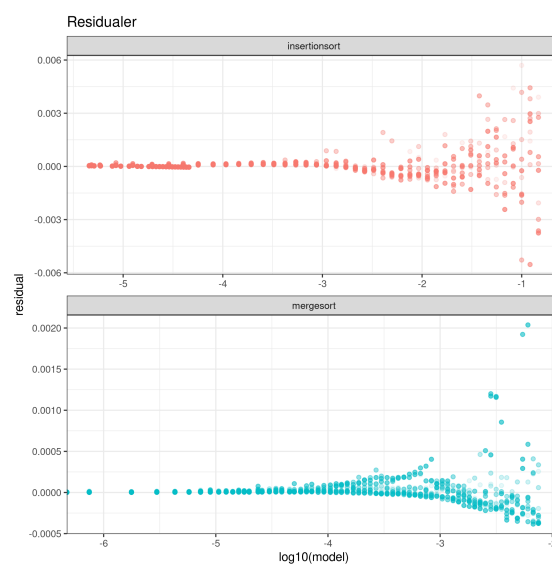
Mergesort implementeret i python [1, s. 106]:

---

```
def mergesort(l):
    if len(l) <= 1:
        return(l)
    else:
        return(merge(mergesort(l[:len(l)//2]), mergesort(l[len(l)//2:])))
```

```
def merge(a,b):
    c = []
    while True:
        if (len(a) == 0):
            return(c + b)
        elif (len(b) == 0):
            return(c + a)
        elif (a[0] <= b[0]):
            c.append(a[0])
            a.pop(0)
        else:
            c.append(b[0])
            b.pop(0)
```

---

(a)  $y = 0.941$ (b)  $y = 3\sin x$ 

Figur 4.1: Sammenligning af insertionsort og mergesort

### 4.3 Sammenligning af Algoritmerne

### 4.4 Den Optimerede Mergesort

## 5 | Den Hurtigste Køretid

Bevis for at den hurtigste køretid for en sorteringsalgoritme er  $n \cdot \log(n)$



## 6 | Konklusion

Her er mine konkluderende sætninger

# Litteraturliste

1. Dietzfelbinger, M. & Mehlhorn, K. *Algoritmer og datastrukturer* <https://github.com/thorehusfeldt/algoritmer-og-datastrukturer/blob/master/ad-book.pdf> ().