Big O notation

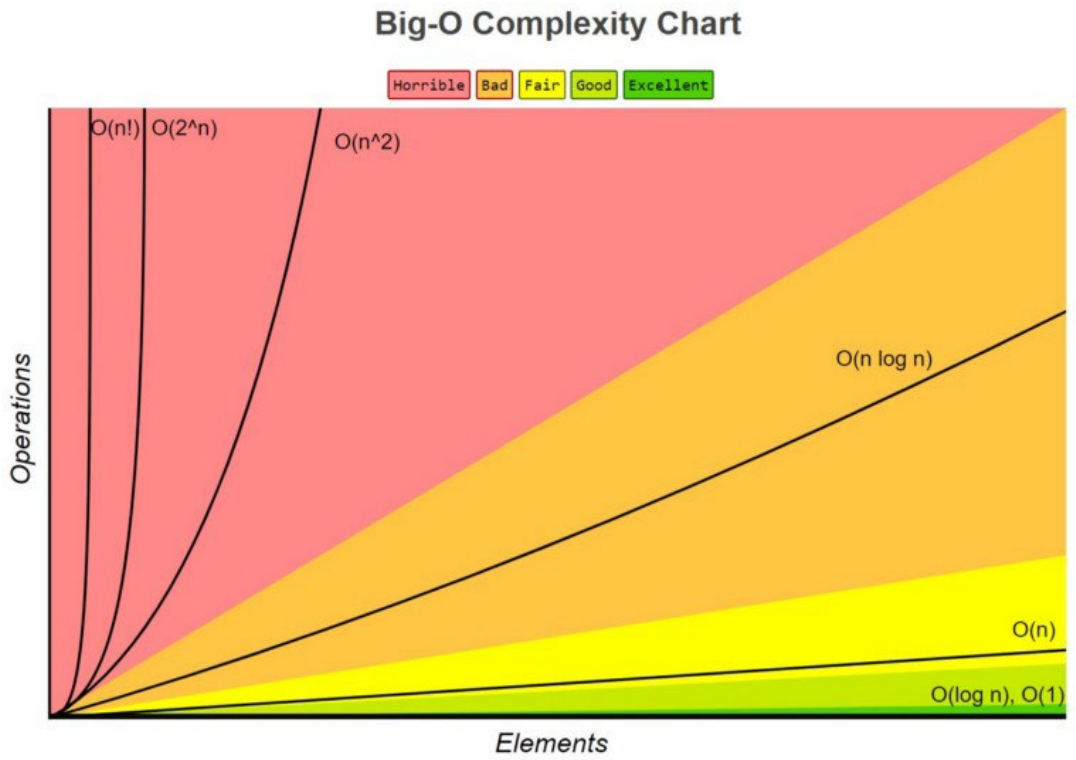
På nearest Neigbour og TSP

Big O notation

- Bruges til at beskrive effektiviteten af en algoritme med hensyn til køretid.
- Bruges til at sammenligne algoritmer med hensyn til vækst.
- Hjælper med at forstå skalerbarheden af algoritmer og forudsige, hvordan de vil fungere, efterhånden som inputstørrelsen vokser.

Notationer

- **O(1)**: Konstant tid
- **O(n)**: Lineær tid
- O(n^2): Kvadratisk tid
- O(log n): Logaritmisk tid
- O(n log n): Lineær-logaritmisk tid

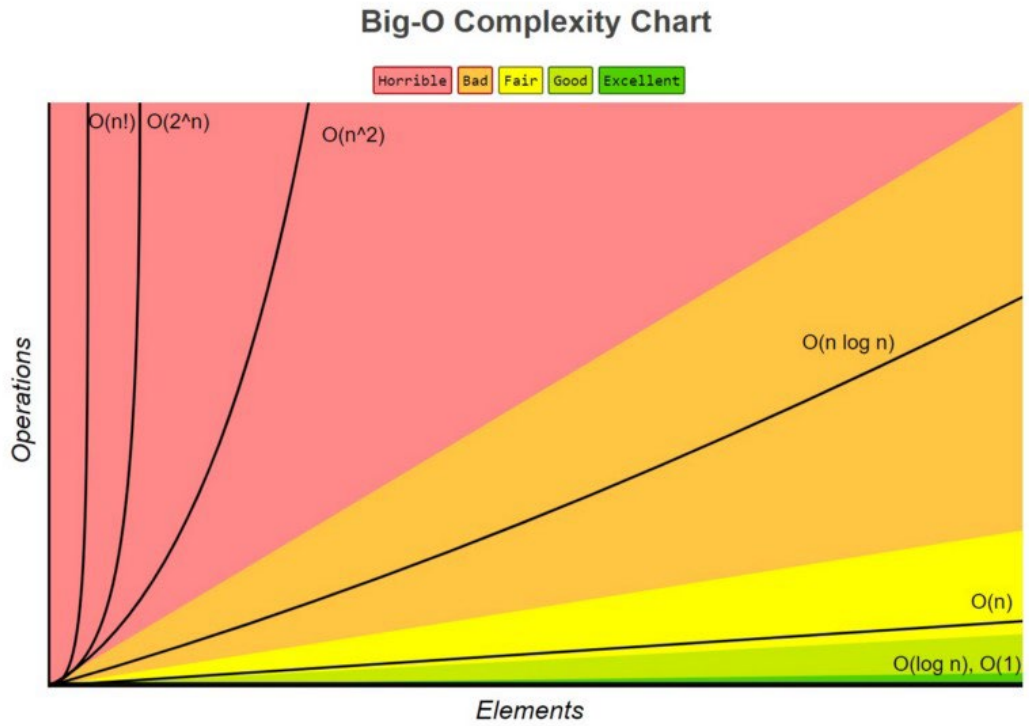


O(1) - Konstant tid

- Definition: Algoritmen tager altid den samme tid, uanset størrelsen på inputtet.
- Egenskab: Uafhængig af, hvor stort inputtet er, forbliver køretiden konstant

Eksempel:

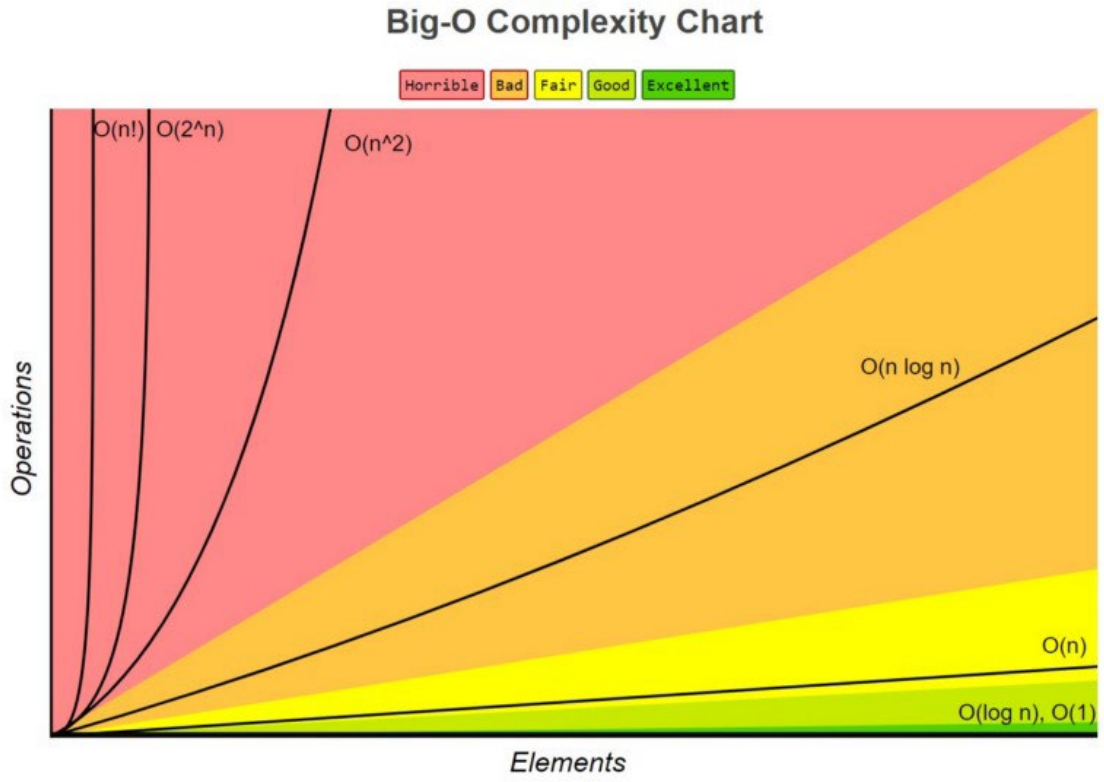
```
int x = array[5];
```



O(n) - Lineær tid

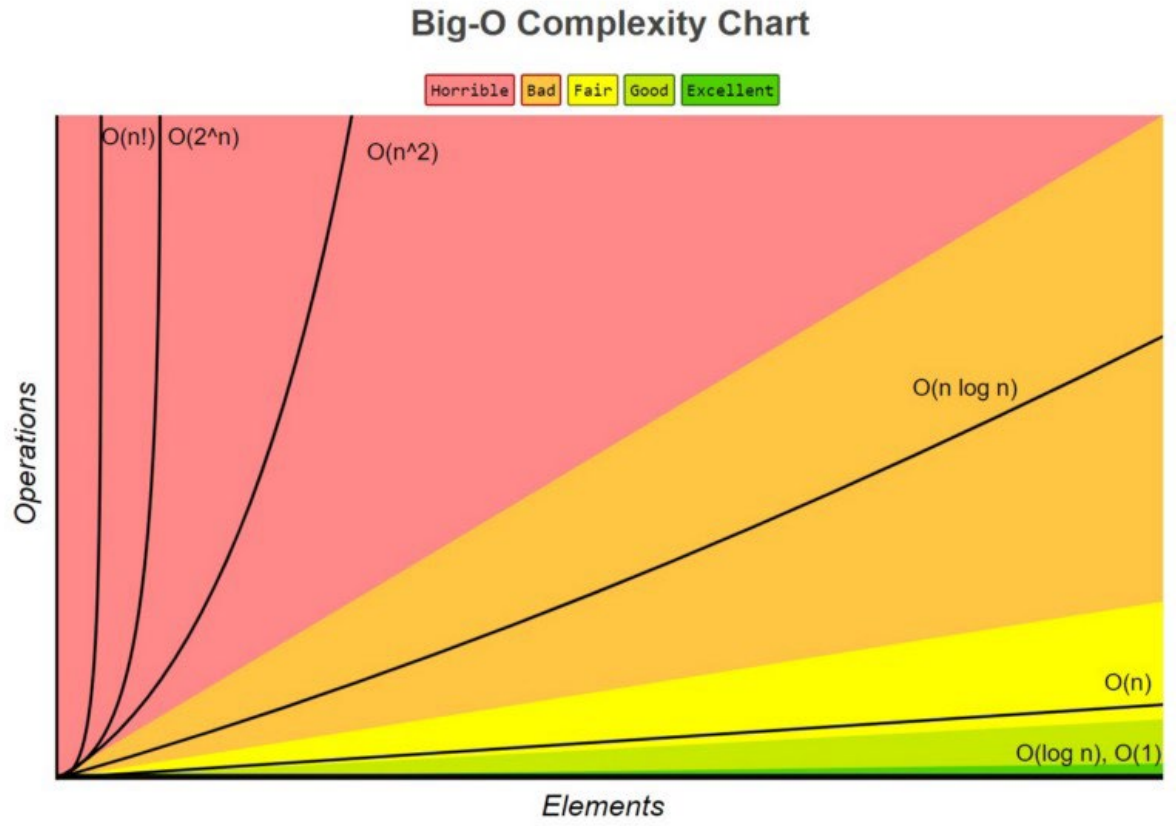
- Algoritmen vokser linært med størrelsen af input. Hvis inputtet fordobles så fordobles køretiden.

```
int[] array = {10, 20, 30, 40};
int n = array.length; // n bliver 4
```



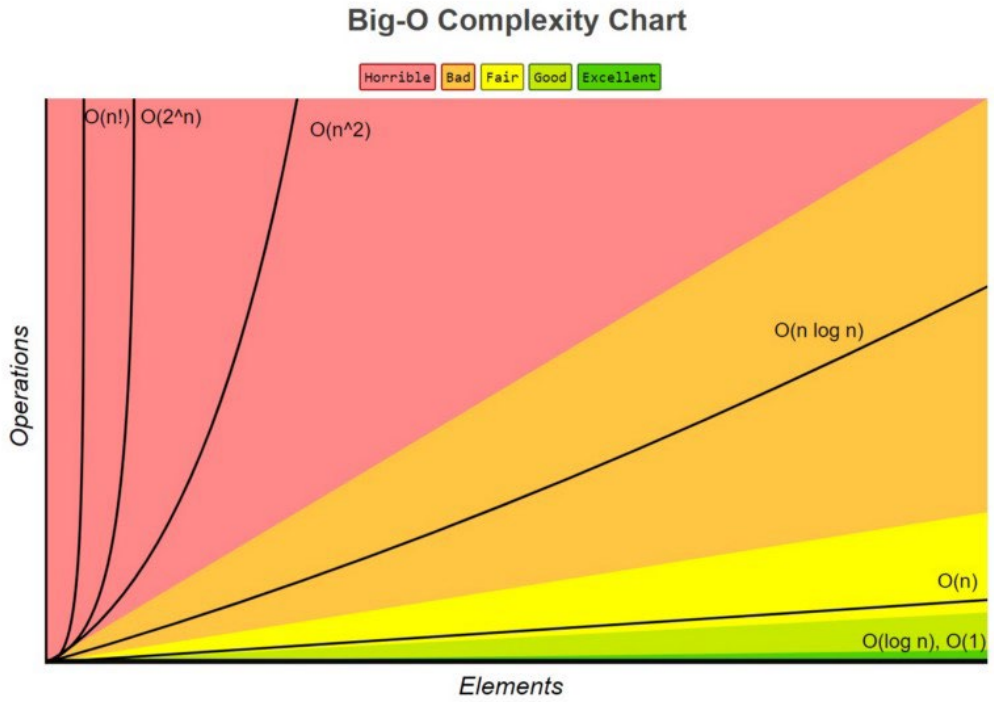
$O(n^2)$ - Kvadratisk tid

- Algoritmen vokser kvadratisk, fx tilføjer vi en by til, så vokser køretiden 4 gange så meget.



O(log n): Logaritmisk tid

- Algoritme tiden vokser langsomt, fordi alt arbejde deles op i iterationer.
- Fx Listen ¨på en binary search: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

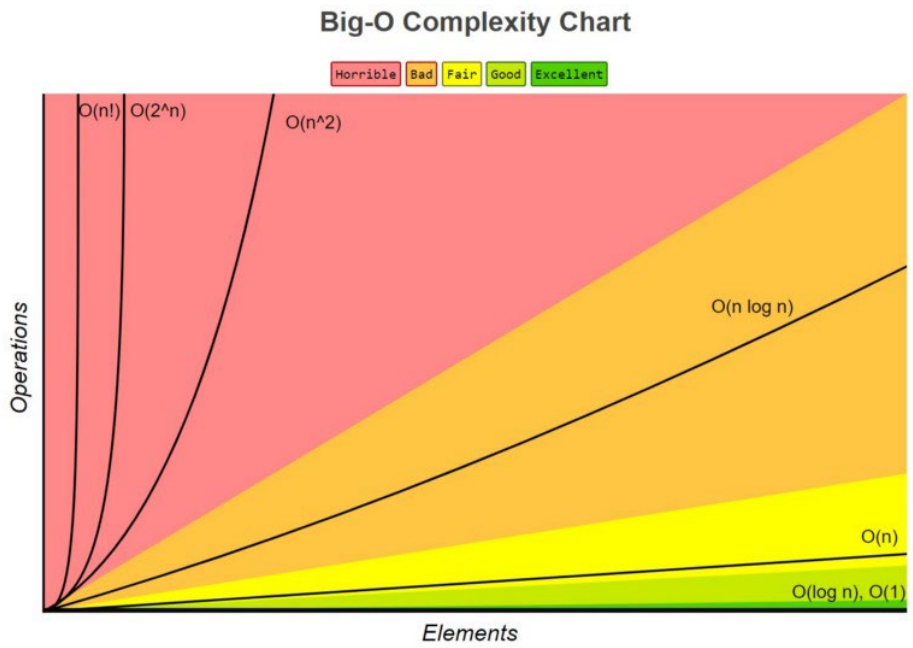


O(n log n): Linær-logaritmisk tid

En kombi af to notationer, som både har linær og logaritmisk.
Typisk ser vi dette notation i algoritmer som Merge sort algoritmer.

Hvis vi bruger eksemplet med en liste på 8 elementer: `[8, 3, 5, 2, 7, 1, 6, 4]`.

- **Opdelingsfasen:**
  - Vi halverer listen $\log_2(8) = 3$ gange:
    - Niveau 1: `[8, 3, 5, 2]` og `[7, 1, 6, 4]`
    - Niveau 2: `[8, 3]`, `[5, 2]`, `[7, 1]`, `[6, 4]`
    - Niveau 3: `[8]`, `[3]`, `[5]`, `[2]`, `[7]`, `[1]`, `[6]`, `[4]`
- **Sammenfletningsfasen:**
  - På hvert niveau skal vi **kombinere og sortere alle elementer én gang**.
    Eksempel:
    - Niveau 3: Sammenflet `[8]` og `[3]` → `[3, 8]` (2 elementer → 2 operationer).
    - Niveau 2: Sammenflet `[3, 8]` og `[2, 5]` → `[2, 3, 5, 8]` (4 elementer → 4 operationer).
    - Niveau 1: Sammenflet `[2, 3, 5, 8]` og `[1, 4, 6, 7]` → `[1, 2, 3, 4, 5, 6, 7, 8]` (8 elementer → 8 operationer).



Big O og nearest Neigbour

$$(n-1)+(n-2)+(n-3)+\cdots+1=\frac{n(n-1)}{2}$$

Afstandsmatrix

Fra/Til	By 0	By 1	By 2	By 3	By 4	By 5
By 0	0	10	15	20	25	30
By 1	10	0	35	25	30	20
By 2	15	35	0	30	20	10
By 3	20	25	30	0	15	25
By 4	25	30	20	15	0	35
By 5	30	20	10	25	35	0

Vi starter fra **by 0**.

Opgave

- Find online eller programmerer selv Nearest Neigbour algoritme.
- udregn antal trin eller samligninger af byer, og se i hvilke notation den er.
- Side Quest (frivilig) find ud af i hvilke notation vil Held karp være i.
- Kan også brug tiden på at lave aflevering.

Links

- https://www.geeksforgeeks.org/analysis-algorithms-big-o-analysis/
- https://www.freecodecamp.org/news/big-o-notation-why-it-matters-and-why-it-doesnt-1674cfa8a23c/