

Big O notation

Jakob Hansen
jakobkha@uio.no

30. august 2021

Big O notasjon

- ▶ Hva er målet med Big O?

Big O notasjon

- ▶ Hva er målet med Big O?
- ▶ Analysere kjøretid! Hvilken algoritme er raskest? (Grovt)

Big O notasjon

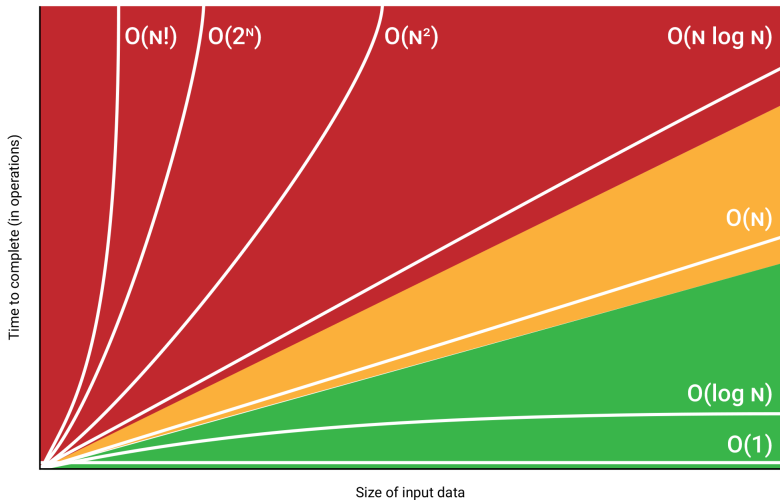
- ▶ Hva er målet med Big O?
- ▶ Analysere kjøretid! Hvilken algoritme er raskest? (Grovt)
- ▶ Abstrahere bort de små forskjellene

Big O notasjon

- ▶ Hva er målet med Big O?
- ▶ Analysere kjøretid! Hvilken algoritme er raskest? (Grovt)
- ▶ Abstrahere bort de små forskjellene

```
1 def numExists(array : list[int], numToFind : int) -> bool:
2
3     # Itererer over array, hvis vi finner tallet, true
4     for i in range(len(array)):
5         if array[i] == numToFind:
6             return true
7
8     # Ingen elementer er lik tallet, false
9     return false
```

Big O notasjon



Konstant tid

- ▶ $O(1)$
- ▶ Tar samme tid uansett
- ▶ Eksempel: Hente første element av et array
- ▶ $O(1000) = O(1)$

Lineær tid

- ▶ $O(n)$
- ▶ Vokser direkte med input størrelse
- ▶ Eksempel: Iterere og printe ut hvert element i et array
- ▶ $O(100n) = O(n)$

Polynomiell tid

- ▶ For hver n , for hver $n \dots$
- ▶ $O(n^x)$, for eksempel $O(n^2)$
- ▶ Eksempel: To løkker som sjekker om det finnes en duplikat i arrayet.
- ▶ Eksempel: “Bruteforce” en kodelås

Logaritmisk tid

- ▶ $O(\log(n))$
- ▶ Litt tricky
- ▶ $\log_2(n) = x$ hvis $2^x = n$
- ▶ Antall ganger du må halvere n for å få 1
- ▶ Eksempel: Lete i telefonbok, Binary search
- ▶ Senere: $O(n \log n)$

Oppgaver!

https://github.com/jakobkhansen/IN2010_h2021