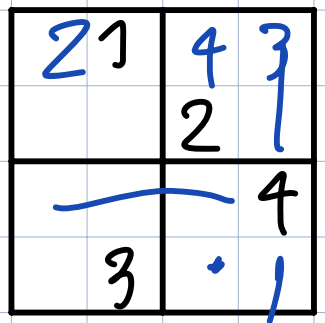
# Velkommen til gruppetime i IN2010

## Uke 10 (?): Kompleksitet

# Plan for timen

- Kompleksitetsklassen P
- Kompleksitetsklassen NP (to definisjoner)
- Reduksjoner
- NP-hard og NP-komplett
- Uberegnbarhet

# Kompleksitetsklasser




$O(n^n)$

$[5, 2, 3, 1]$ $O(n \cdot \log n)$

# Kompleksitetsklassen P

- klassen av alle problemer som kan løses i polynomiell tid

$O(n^k)$ hvor k er konstant

f.eks $O(n^2)$, $O(n^3)$, $O(n)$, $O(1)$, $O(n^{10000})$

$O(n \cdot \log n)$ er også i P

$O(n^2) \geq O(n \cdot \log n)$

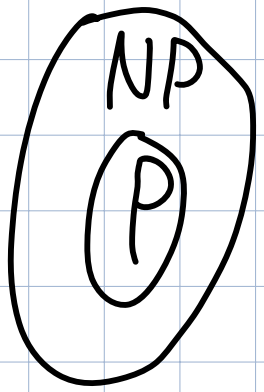
$O(2^n)$ er ikke i P



# Kompleksitetsklassen NP

- Klassen av alle problemer som kan ~~løses~~ verifiseres i polynomiell tid

- f.eks. Løse sudoku

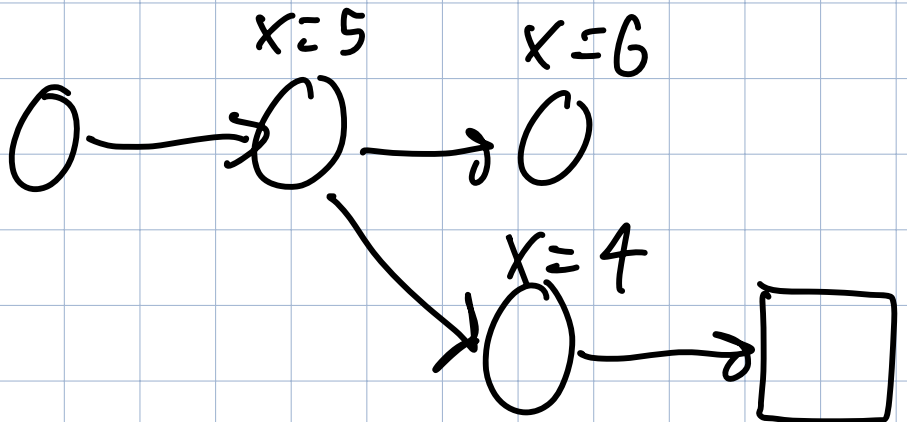


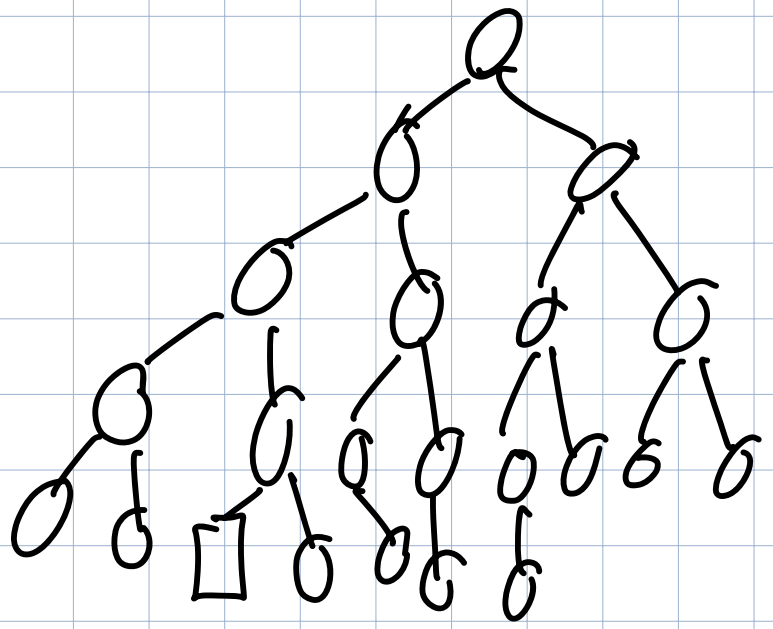

```
def sjekkSudoku(sudoku):
    for hver rad:
        sjekk om den inneholder
        hvert tall 1 gang, return false hvis ikke
    for hver kolonne:
        —— " ——
    for hver boks:
        —— " ——
    return true
```

$O(n^2)$

# Kompleksitetsklassen NP

- klassen av alle problemer som kan løses i ikke-deterministisk polynomiell tid



x = 5

x += 1 eller x -= 1

```
if x=4:
    return True
```

# Kompleksitetsklassen NP

```
def solveSudoku(sudoku):
    for every empty slot:
        num ∈ velg et tall fra 1 til n
        put num in slot
    if sjekkSudoku(sudoku):
        return sudoku
```

Ikke-deterministisk $O(n^2)$ $\rightarrow$ Sudoku er i NP

# Reduksjoner

A er "lettere" enn B hvis A kan reduseres til B
(eller like vanskelig)
i polynomiell tid

# NP-harde problemer

- Et problem A er NP-hardt hvis

  alle problemer i NP kan reduseres til A.

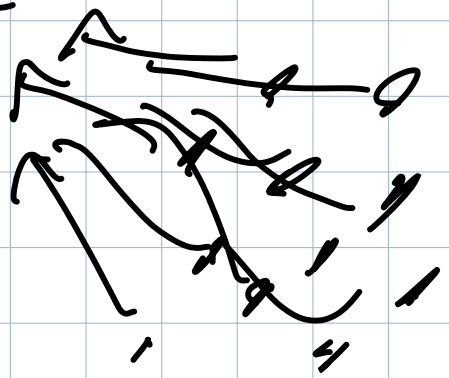
  (dvs. at alt i NP er "lettere eller like vanskelig" som A)

- SAT er NP-hardt

- Siden SAT kan reduseres til sudoku,
  så kan alt i NP reduseres til sudoku (via SAT),
  dermed er sudoku NP-hardt

# NP-harde problemer

sudoku ← SAT



↓

stoppeproblemet

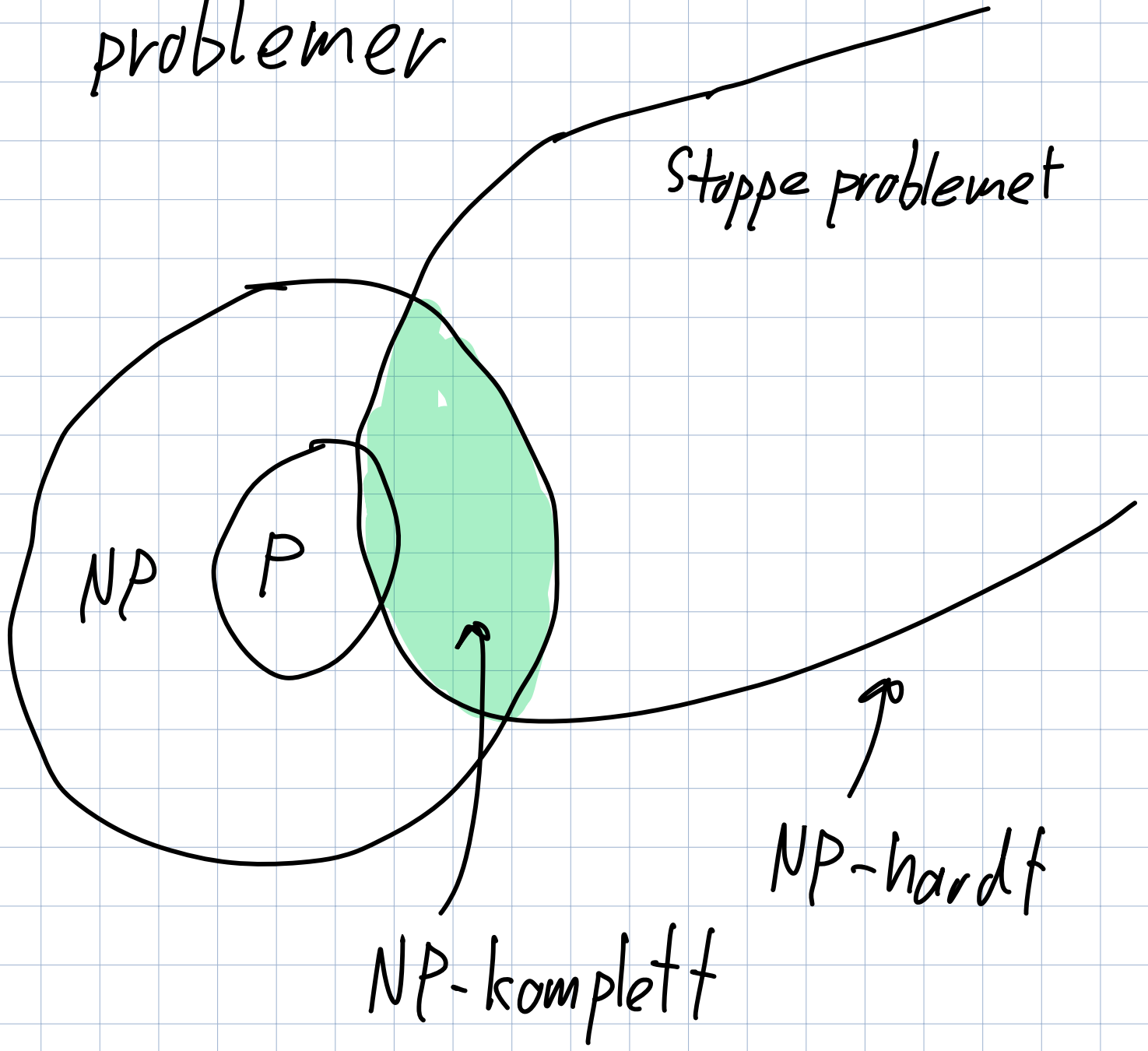
↑

Uberegnbart
Ikke i NP
Det er NP-hardt

# NP-komplette problemer

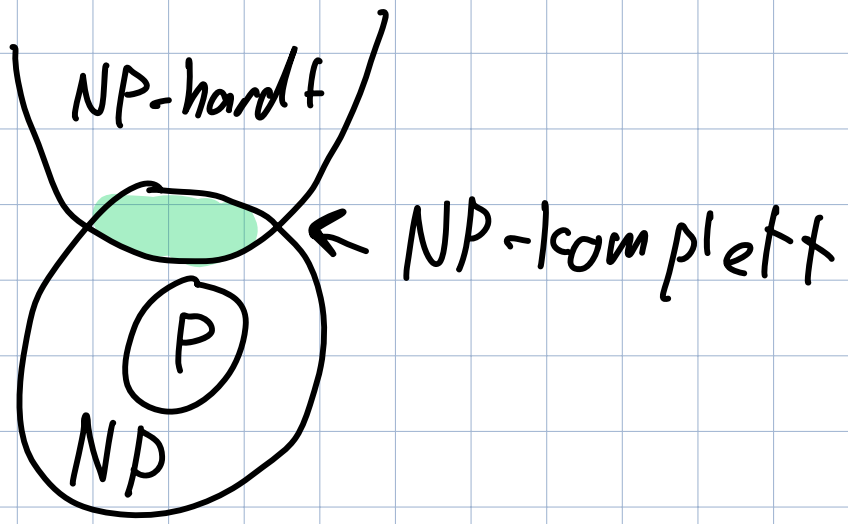
- Problemer som er i NP, og som er NP-harde
- De vanskeligste problemene i NP
- F.eks
  - Sudoku
  - SAT
  - Travelling salesperson (TSP)
- Ikke NP-komplett, men NP-hardt (altså ikke i NP)
  - Stoppeproblemet (halting problem)
  - Å finne det beste trekket i sjakk

# NP-komplette problemer

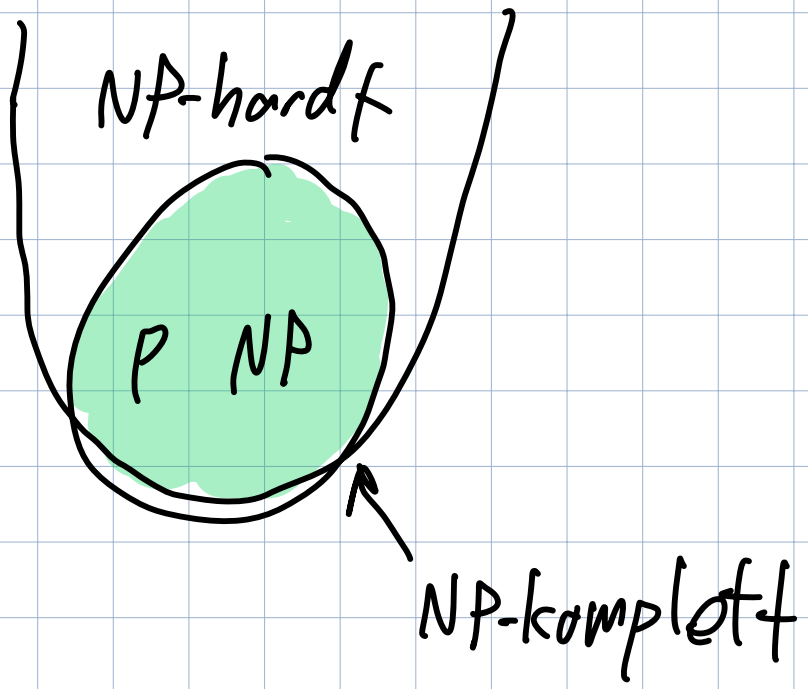


# NP-komplette problemer

Hvis $P \neq NP$




Hvis $P = NP$



# NP-komplette problemer

