

:: Forside

Introduksjon

Fibonacci-tall

Memoisering

DP

Neste uke

Spørsmål

Introduksjon til memoisering og dynamisk programmering

Åsmund Eldhuset

[asmunde *at* stud.ntnu.no](mailto:asmunde*at*stud.ntnu.no)

folk.ntnu.no/asmunde/algdat/mem-dp-intro.ppt

Forside

:: Introduksjon

Fibonacci-tall

Memoisering

DP

Neste uke

Spørsmål

Memoisering og DP

- Dette er to ganske like teknikker for å lage algoritmer
- De kan brukes på svært mange tilsynelatende forskjellige problemer
- Kan gjøre eksponentiell kjøretid om til polynomisk kjøretid
- Vi vil nå "finne opp" memoisering og DP mens vi løser et problem

Forside

Introduksjon

:: Fibonacci-tall

Memoisering

DP

Neste uke

Spørsmål

Fibonacci-tall

- Problem: Regn ut Fibonacci-tall nummer n
- Matematisk definisjon:
 - $f(0) = 0$
 - $f(1) = 1$
 - $f(n) = f(n - 1) + f(n - 2)$ for $n > 1$
- Hvert tall er altså summen av de to foregående: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55...

Forside

Introduksjon

:: Fibonacci-tall

Memoisering

DP

Neste uke

Spørsmål

Fibonacci-tall

- Dette ser lett ut å programmere. Vi fyrer løs med følgende Python-kode:
- ```
def f(n):
 if n <= 0:
 return 0
 elif n == 1:
 return 1
 else:
 return f(n - 1) + f(n - 2)
```

Forside

Introduksjon

:: Fibonacci-tall

Memoisering

DP

Neste uke

Spørsmål

# Fibonacci-tall

- Dessverre blir kjøretiden helt forferdelig!
  - For  $n < 20$  går det bra
  - Utregning av  $f(25)$  tar et halvt sekund
  - $f(30)$  tar fem sekunder
  - $f(35)$  tar et minutt
  - $f(42)$  vil vare ut forelesningen
  - $f(50)$  vil ta et år
- Når  $n$  øker med 1, dobles nesten kjøretiden
- Hva skyldes dette?
- La oss ta en kikk på hva funksjonen gjør



Forside

Introduksjon

Fibonacci-tall

:: Memoisering

DP

Neste uke

Spørsmål

# Memoisering

- Løsning: Når funksjonen har regnet ut et Fibonacci-tall, registrerer vi resultatet
- Når funksjonen blir bedt om å regne ut et Fibonacci-tall, sjekker vi først om vi allerede har regnet det ut
  - Har vi allerede regnet det ut, returnerer vi resultatet vi har
  - Ellers starter vi utregningen på vanlig måte
- Kan bruke enten dictionary eller array
- Array er raskere, men du må vite størrelsen på forhånd

Forside

Introduksjon

Fibonacci-tall

:: Memoisering

DP

Neste uke

Spørsmål

# Memoisering med dictionary

```
def f(n, dict):
 if n <= 0:
 return 0
 elif n == 1:
 return 1
 elif dict.has_key(n):
 return dict[n]
 else:
 result = f(n - 1, dict) + f(n - 2, dict)
 dict[n] = result
 return result

def fib(n):
 dict = {}
 return f(n, dict)
```

Forside

Introduksjon

Fibonacci-tall

:: Memoisering

DP

Neste uke

Spørsmål

# Memoisering med array

```
def f(n, array):
 if n <= 0:
 return 0
 elif n == 1:
 return 1
 elif array[n] != -1:
 return array[n]
 else:
 result = f(n - 1, array) \
 + f(n - 2, array)
 array[n] = result
 return result

def fib(n):
 array = [-1] * (n + 1)
 return f(n, array)
```

Forside

Introduksjon

Fibonacci-tall

Memoisering

:: DP

Neste uke

Spørsmål

# Dynamisk programmering

- Vi ser at utregningen av hvert tall bare avhenger av tidligere tall
- Derfor kan vi faktisk fylle ut arrayet vårt direkte
- Går raskere enn memoisering pga. ingen rekursjon

Forside

Introduksjon

Fibonacci-tall

Memoisering

:: DP

Neste uke

Spørsmål

# Dynamisk programmering

```
def f(n):
 if n <= 0:
 return 0
 array = [-1] * (n + 1)
 array[0] = 0
 array[1] = 1
 for i in range(2, n + 1):
 array[i] = array[i - 1] + array[i - 2]
 return array[n]
```

Forside

Introduksjon

Fibonacci-tall

Memoisering

:: DP

Neste uke

Spørsmål

# DP – plassoptimalisering

- Vi ser at utregningen av hvert tall bare avhenger av de to forrige tallene
- Dermed klarer vi oss med bare å lagre tre tall av gangen – sparer minne
- Merk at dette bare er nyttig hvis man kun er interessert i det siste tallet, og ikke vil beholde tallene underveis

Forside

Introduksjon

Fibonacci-tall

Memoisering

:: DP

Neste uke

Spørsmål

# DP – plassoptimalisering

```
def f(n):
 if n <= 0:
 return 0
 previous = 0
 current = 1
 for i in range(n - 1):
 next = previous + current
 previous = current
 current = next
 return current
```

Forside

Introduksjon

Fibonacci-tall

Memoisering

DP

:: Neste uke

Spørsmål

# I neste øvingsforelesning...

- Når kan vi bruke memoisering og DP?
- Generell tankegang
- Mange eksempler på DP
- Sammenligning med grådige algoritmer
- Neste øvingsforelesning forutsetter at man kan det som står i denne foilen!

Forside

Introduksjon

Fibonacci-tall

Memoisering

DP

Neste uke

:: Spørsmål

# Spørsmål?

---