

Imperativ Programmering og Datastrukturer

Rekursion

René Rydhof Hansen

29. oktober 2008

Mål

- At kunne forklare/definere hvad rekursion er
- At kunne implementere givne rekursive funktioner
- At kunne forklare hvordan rekursion kan bruges som problemløsningsstrategi

For at forstå rekursion... skal man først forstå rekursion

Rekursion

- Rekursion er en problemløsningsstrategi baseret på løsning af delproblemer *af samme slags* som hovedproblemet.
- Der skal være et *basistilfælde* der ikke involverer rekursion for at sikre terminering.

Example

En folder kan indeholde

- Filer
-

Rekursion vs. Iteration

- Løkker kan altid formuleres som rekursion
- Rekursion kan, med noget besvær, formuleres som iteration

For at forstå rekursion... skal man først forstå rekursion

Rekursion

- Rekursion er en problemløsningsstrategi baseret på løsning af delproblemer *af samme slags* som hovedproblemet.
- Der skal være et *basistilfælde* der ikke involverer rekursion for at sikre terminering.

Example

En folder kan indeholde

- Filer
-

Rekursion vs. Iteration

- Løkker kan altid formuleres som rekursion
- Rekursion kan, med noget besvær, formuleres som iteration

For at forstå rekursion... skal man først forstå rekursion

Rekursion

- Rekursion er en problemløsningsstrategi baseret på løsning af delproblemer *af samme slags* som hovedproblemet.
- Der skal være et *basistilfælde* der ikke involverer rekursion for at sikre terminering.

Example

En folder kan indeholde

- Filer
- ... og *andre foldere*

Rekursion vs. Iteration

- Løkker kan altid formuleres som rekursion
- Rekursion kan, med noget besvær, formuleres som iteration

Example

Fakultetsfunktionen (!):

$$n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1 = \begin{cases} 1 & n = 0 \\ n \cdot (n - 1)! & \text{ellers} \end{cases}$$

```
public static int fact(int i) {  
    if (i == 0) {  
        return 1;  
    } else {  
        return i * fact(i-1);  
    }  
}
```

Endnu et eksempel: Fibonacci-tallene

Ballade på Kaninbjerget

Placér et kaninpar i et bur. Under antagelse af, at et kaninpar først kan producere et nyt kaninpar efter to måneder og derefter producerer et kaninpar hver måned, hvor mange kaninpar er der produceret efter et år?

Svar

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

Definition

$$\text{fib}(n) = \begin{cases} 0 & \text{hvis } n = 0 \\ 1 & \text{hvis } n = 1 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{ellers} \end{cases}$$

Endnu et eksempel: Fibonacci-tallene

Ballade på Kaninbjerget

Placér et kaninpar i et bur. Under antagelse af, at et kaninpar først kan producere et nyt kaninpar efter to måneder og derefter producerer et kaninpar hver måned, hvor mange kaninpar er der produceret efter et år?

Svar

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

Definition

$$\text{fib}(n) = \begin{cases} 0 & \text{hvis } n = 0 \\ 1 & \text{hvis } n = 1 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{ellers} \end{cases}$$

Endnu et eksempel: Fibonacci-tallene

Ballade på Kaninbjerget

Placér et kaninpar i et bur. Under antagelse af, at et kaninpar først kan producere et nyt kaninpar efter to måneder og derefter producerer et kaninpar hver måned, hvor mange kaninpar er der produceret efter et år?

Svar

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

Definition

$$\text{fib}(n) = \begin{cases} 0 & \text{hvis } n = 0 \\ 1 & \text{hvis } n = 1 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{ellers} \end{cases}$$

Fibonacci-tallene i C#

```
public static int fib(int n) {  
    if(n == 0) {  
        return 0;  
    } else if (n == 1) {  
        return 1;  
    } else {  
        return fib(n-1) * fib(n-2);  
    }  
}
```

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

```
x
xxx
xxxxx
0      1      2
```

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

```
xxx
xxxxx      x
0          1  2
```

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

xxxxx	xxx	x
0	1	2

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

	x	
xxxxx	xxx	
0	1	2

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

	x	
	xxx	xxxxx
0	1	2

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

x	xxx	xxxxx
0	1	2

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

		xxx
x		xxxxx
0	1	2

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

			x
			xxx
			xxxxx
0	1	2	

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Rekursion som generel strategi: Hanoi's Tårne

En gang for længe siden...

- Skiverne skal flyttes fra pæl 0 til pæl 2
- En skive kan flyttes til en tom pæl
- En skive kan flyttes over på en *større* skive

		x
		xxx
		xxxxx
0	1	2

Rekursiv Løsning

- Flyt øverste 2 skiver fra 0 til 1 via 2
- Flyt sidste skive direkte fra 0 til 2
- Flyt øverste 2 skiver fra 1 til 2 via 0

Opsamling og Repetition

- Opsamling på programmeringsdel
- Indledning til datastrukturer
- Repetitionsemner (sendes til `rrh@cs.aau.dk`)