# Introduktion til Programmering og Problemløsning (PoP)

Jon Sporring
Department of Computer Science
2022/09/02

UNIVERSITY OF COPENHAGEN

# F# er en lommeregner

```
357
864
357+864
```

```
let a = 357
let b = 864
let c = a + b
do printfn "%A" c
```

3 måder at køre (execute / run) programmet på:

- dotnet fsi -> indtast myFirstFsharp.fsx
- dotnet fsi myFirstFsharp.fsx
- dotnet run myFirstFsharp.fsx

# Nedtælling

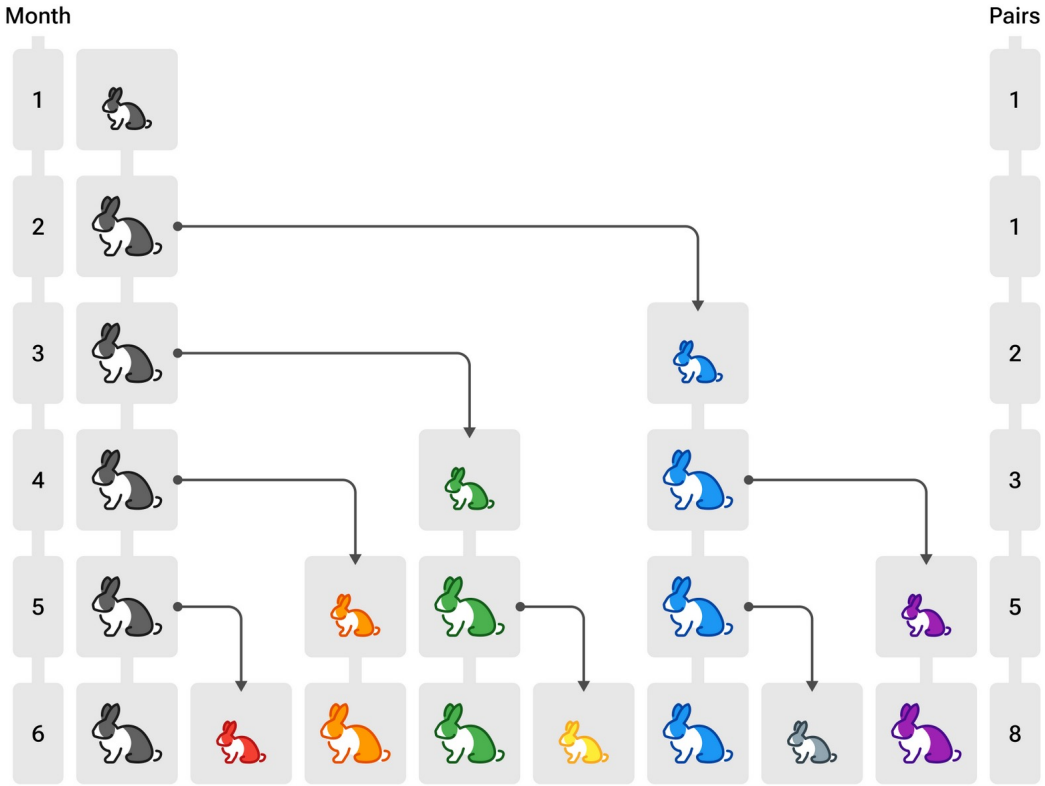
```
let rec countDown n =
    printfn "%A" n
    match n with
      0 -> ()
    | _ -> countDown (n-1)

countDown 5
```

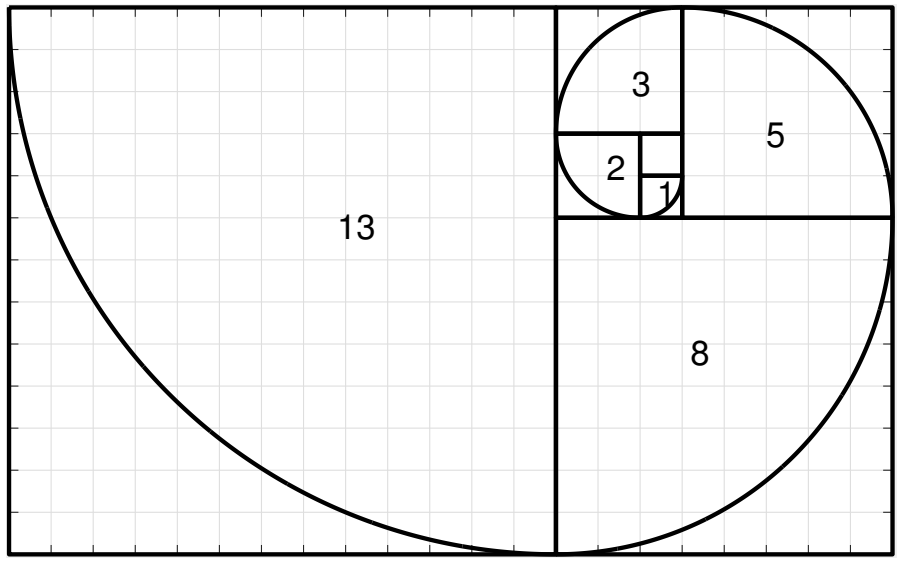
```
let countDown n =
    let mutable i = n
    while i >= 0 do
      printfn "%A" i
      i <- i - 1

countDown 5
```

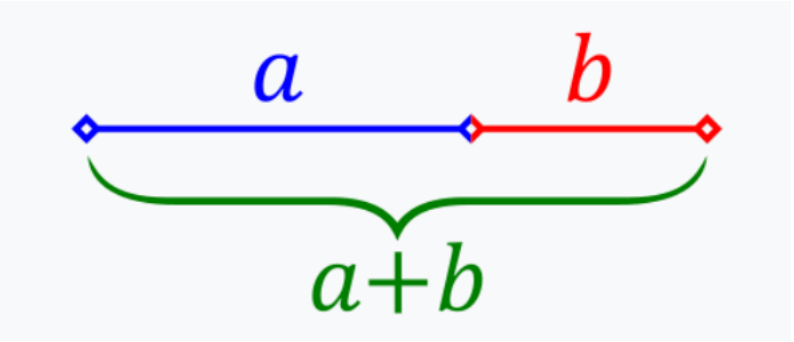
3 måder at køre (execute / run) programmet på:

- dotnet fsi -> indtast myFirstFsharp.fsx
- dotnet fsi myFirstFsharp.fsx
- dotnet run myFirstFsharp.fsx

# Fibonacci: 0 1 1 2 3 5 8 13 21 …




$$\frac{a+b}{a} = \frac{a}{b}$$

$$1 + \frac{b}{a} = \frac{a}{b}$$

$$\phi = \frac{a}{b}$$

$$1 + \frac{1}{\phi} = \phi$$

$$\phi + 1 = \phi^2$$

$$\phi = \frac{1 \pm \sqrt{5}}{2}$$

$$= (-0.618\,,\, 1.618)$$

$$F(0) = 0$$

$$F(1) = 1$$

$$F(n) = F(n-1) + F(n-2)$$

# Fibonacci: 0 1 1 2 3 5 8 13 21 ⋯

## Recursive

```
let rec fib n =
  match n with
    0 | 1 ->
      1
    | _ ->
      fib (n - 1) + fib (n - 2)

for i = 0 to 45 do
  printfn "fib(%d) = %d" i (fib i)
```

## Imperative

```
let fib n =
  match n with
    0 | 1 ->
      1
    | _ ->
      let mutable prevPrev = 0
      let mutable prev = 1

      for i = 2 to n do
        let curr = prev + prevPrev
        prevPrev <- prev
        prev <- curr
      prev

for i = 0 to 45 do
  printfn "fib(%d) = %d" i (fib i)
```

```
% time dotnet fsi fibRecursive.fsx
fib(0) = 1
fib(1) = 1
fib(2) = 2
fib(3) = 3
fib(4) = 5
fib(5) = 8
…
fib(43) = 701408733
fib(44) = 1134903170
fib(45) = 1836311903

dotnet fsi fibRecursive.fsx  14.34s user ⋯

% time dotnet fsi fibImperative.fsx
…
fib(44) = 701408733
fib(45) = 1134903170
dotnet fsi fibImperative.fsx  0.98s user ⋯
```

# Rekursion

$$F(0) = 0$$

$$F(1) = 1$$

$$F(n) = F(n-1)+F(n-2)$$

Fib(5)

= fib(4) + fib(3)

= (fib(3)+fib(2)) + (fib(2)+fib(1))

= ((fib(2)+fib(1)) + (fib(1)+fib(0)))+((fib(1)+fib(0))+1)

= (((fib(1)+fib(0))+1) + (1+0))+((1+0)+1)

= (((1+0)+1) + (1+0))+((1+0)+1)

= ((1+1) + 1)+(1+1)

= (2 + 1)+2

= 3+2

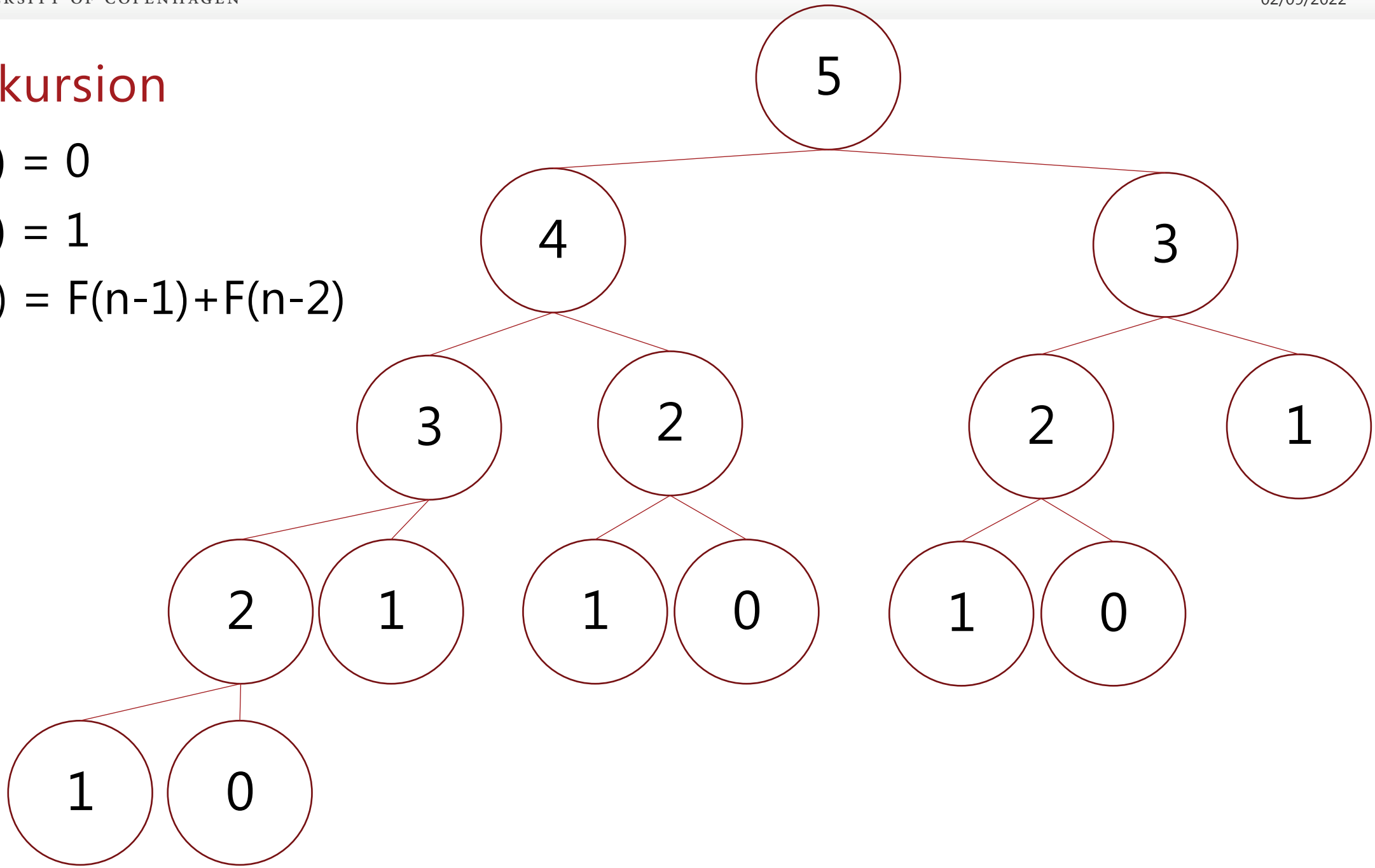
= 5

# Rekursion

F(0) = 0

F(1) = 1

F(n) = F(n-1)+F(n-2)



# Resumé

Denne video fortalte om:

- F# som en lommeregner
- Konstanter, typer, bindinger, betingelser, løkker, printfn
- Fibonaccis talrække