4.3: Rekursivt definerede funktioner (følger).

For at definere en uendelig følge

$$f(0), f(1), f(2), \ldots$$

skal vi

Basisskridt: angive en værdi for $f(0)$

Rekursionsskridt: for ethvert $n \geq 0$ angive hvordan man bestemmer $f(n+1)$ fra $f(0), \ldots, f(n)$.

Man kan eventuelt ændre alle røde 0'er til et andet tal b.

Fibonaccitallene $f_0, f_1, f_2, \ldots$ defines rekusivt ved

Basisskridt: $f_0 = 0$

Rekursionsskridt:

$f_1 = 1$

for $n \geq 2$ er $f_n = f_{n-2} + f_{n-1}$

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots$$

Sætning

$$f_n = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right)$$

4.2: Stærk induktion.

For $n \in \mathbb{Z}$ lad $P(n)$ betegne et udsagn, der kan være sandt eller falsk. Sandhedsværdien kan være forskellig for forskellige værdier af n.

Induktionsprincippet: For at bevise at $P(n)$ er sand for alle $n \geq 1$ skal vi
Basisskridt: bevise at $P(1)$ er sand.
Induktionsskridt: bevise at der for ethvert $k \geq 1$ gælder: hvis $P(1), \ldots, P(k)$ alle er sande så er $P(k+1)$ også sand.

Man kan eventuelt ændre alle røde 1-taller til et andet tal b. Det grønne 1-tal må ikke ændres.

4.2: Velordningsprincippet.

Lad S være en *ikke-tom* mængde af ikke-negative hele tal.
Så har S et mindste element, altså et element $m \in S$ så $s \geq m$ for alle $s \in S$.

Anvendelse: $P(n)$ et udsagn. Vi skal bevise at $P(n)$ er sand for alle $n \geq 0$.
Lad $S = \{n \text{ ikke-negativ heltal} \mid P(n) \text{ er falsk}\}$.
Skal vise: $S = \emptyset$.
Bevis ved modstrid: antag $S \neq \emptyset$ og lad $m \in S$ være det mindste element. ...

4.4: Rekursive algoritmer.

Algoritme 4: Beregn største fælles divisor Side 313

procedure gcd (a, b: ikke-negative heltal, $a < b$)
if $a = 0$ **then** gcd(a, b) := b
else gcd(a, b) := gcd(b mod a, a)

procedure gcd(a, b: positive heltal)

x:=a,

y:=b

while $y \neq 0$

begin

$r := x \mod y$

$x := y$

$y := r$

end

procedure fibonacci (n: ikke-negativt heltal)
if $n = 0$ **then** fibonacci(0) := 0
else
    if $n = 1$ **then** fibonacci(1) := 1
    else fibonacci(n) := fibonacci($n-1$)+fibonacci($n-2$)

```
procedure fibonacci (n: ikke-negativt heltal)
if n = 0 then  y:=0
else
begin
    x := 0
    y := 1
    for i := 1 to n − 1
    begin
        z := x + y
        x := y
        y := z
    end
end
{ y = f_n }
```