

---

## DD2458, Problemlösning och programmering under press

### Föreläsning 0: Introduktion och administration

Datum: 2006-08-31

Skribent(er): Osquarulda Teknolog

Föreläsare: Mikael Goldmann

---

Den här föreläsningen behandlar hur värdfullt det är att noga repetera saker för att man ska lära sig dem ordentligt. Den här föreläsningen behandlar hur värdfullt det är att noga repetera saker för att man ska lära sig dem ordentligt.

## 1 En rubrik

Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera.

Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera.

**Definition 1.1** *Låt  $\mathbf{Z}$  beteckna heltalen.*

**Definition 1.2** *Låt  $\mathbf{N} = \{n \geq 0 \mid n \in \mathbf{Z}\}$  vara de naturliga talen.*

### 1.1 En underrubrik

Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera. Det finns mycket att säga som är värt att repetera.

## 2 En rubrik till

Repetera, repetera och repetera.

Repetera, repetera och repetera.

**Lemma 2.1** *En matematisk sats:*

$$\forall x : x \in \mathbf{Z} \Rightarrow x^2 \in \mathbf{N}$$

**Sats 2.2** *En matematisk sats:*

$$\forall x : A(x) \Rightarrow B(X)$$

### 3 En algoritm

Med hjälp av `algorithm`-omgivningen kan man typsätta pseudokod ganska snyggt.

**Algoritm 1:** Ett exempel på `algorithm`-omgivningen.

**Input:** En icke-tom mängd  $U = \{u_1, u_2, \dots, u_n\}$  av heltal

**Output:** Det största elementet och en mängd  $A$ ,  $|A| = \log n$ , som innehåller det näst största elementet. (Om  $n = 1$ , så  $A = \emptyset$ ).

`MAX2(U)`

```
(1)  if |U| = 1
(2)      return (u1, ∅)
(3)  else if |U| = 2
(4)      if u1 > u2 then return (u1, {u2})
(5)          else return (u2, {u1})
(6)  else
(7)      (b, B) ← MAX2({ui}i=1⌊n/2⌋)
(8)      (c, C) ← MAX2({ui}i=⌊n/2⌋+1n)
(9)      if b > c then return (b, {c} ∪ B)
(10)     else return (c, {b} ∪ C)
```