```
for i=0 to n {
    for j=1 to n {
        for k=1 to 4 {
            method(i,j,k);
        }
    }
}
```

= konstant

= avhengig av instans

$(n+1) \cdot n \cdot 4 \cdot \text{Tid}(\text{method})$

$O(n^2) \cdot \text{Tid}(\text{method})$

```
recursiveMax(A, n) {
      if n=1 then
         return A[0]
      return max(A[n-1], recursiveMax(A,n-1))
   }
```

}3 3

1 1 1 1 1 1

= konstant

= avhengig av instans

$$f(n) = \begin{cases} 3 & (n=1) \\ 7 + f(n-1) & (n>1) \end{cases}$$

$$\begin{aligned} f(n) &= 7 + f(n-1) \\ &= 7 + (7 + f(n-2)) \\ &= 7 + 7 + (7 + f(n-3)) \\ &\vdots \\ &= 7 + \dots + 7 + f(1) \\ &= \underbrace{7 + \dots + 7}_{n-1} + 3 = 7 \cdot (n-1) + 3 = 7n - 4 \end{aligned}$$

= konstant

= avhengig av instans

```
BinarySearch(A[0..N-1], value, low, high) {
      if (high < low)
          return not_found
      mid = (low + high) / 2
      if (A[mid] > value)
          return BinarySearch(A, value, low, mid-1)
      else if (A[mid] < value)
          return BinarySearch(A, value, mid+1, high)
      else
          return mid
  }
```

$\}\ c$

$$f(n) = c + f\left(\frac{n}{2}\right)$$
$$= c + c + f\left(\frac{n}{4}\right)$$
$$= c + c + c + f\left(\frac{n}{8}\right)$$
$$\vdots$$
$$= c + \dots + c + f(1)$$
$$= \underbrace{c + \dots + c + c}_{\log_2(n)}$$
$$= c \cdot \log_2(n)$$

$$2^{\log_2(n)} = n$$

"Hvor mange ganger å gange 2 med seg selv for å få n".

$$\boxed{\text{"Hvor ofte dele } n \text{ på 2 for å få 1".}}$$