

CoMa Übung XIII, 05.02.13

1. Katayoun Chaman Ara
2. Debora Wetzel
3. Dennis Wenzel

Mi, Übung:

1. Timo Ayush
2. Alejandro Torres Orjuela
3. Stefan Hildebrand

Mi, Sprechstunde:

1. Kevin Stiefel
2. Qiao Luo
3. Svenja Großholz
4. Tobias Jesche

Hallo

$$f \in \Omega(g) \iff \exists c > 0, n_0 \in \mathbb{N}, f(n) \geq c g(n), \forall n \geq n_0$$

$$\geq c g(n), \forall n \geq n_0$$

$$f(n) = n^2 + \sqrt{n} + \log n$$

$$\in \Omega(n^2), \in O(n^2) \implies \in \Theta(n^2)$$

```

public int binarySearch(int[] array, int value) {
    int m = array.length / 2; int l = 0; int r = array.length - 1;
    while (l <= r) {
        if (value == array[m]) {
            return m;
        } else if (value > array[m]) {
            l = m + 1;
        } else {
            r = m - 1;
        }
    }
}

```

$$f(n) = f(n-1)^3 \quad f(1) = 3$$

```
public int doSomething (int n) {  
    if (n == 1) return 3;  
    else {  
        return Math.paldoSomething (n-1, 3);  
    }  
}
```

```
public class Student {  
    private int number;  
    public Student (int number) {  
        this.number = number;  
    }  
}
```

```

    }
    public int getNumber () {
        return this.number;
    }
}

public class CoMaStudent extends Student
    implements Comparable<CoMaStudent> {
    private int punkte;

    public CoMaStudent (int punkte, int MN.)
    {
        super (MN.);
        this.punkte = punkte;
    }

    public int compareTo<CoMaStudent>
}

```

super Kati ☺

```
public class Rectangle {
```

```
    private double length;  
    private double width;
```

```
    public Rectangle () {
```

```
        this.length = 1;
```

```
        this.width = 1;
```

```
    }
```

```
    public Rectangle (double length, double width) {
```

```
        this.length = length;
```

```
        this.width = width;
```

```
    }
```

```
    public double area() {
```

```
        return this.length * this.width;
```

```
    }
```

```
}
```

```
public class Square extends
```

```
    implements HasArea Rectangle {
```

```
    public Square (double length) {
```

```
        super (length, length);
```

```
    }
```

```
}
```

```
public interface HasArea {
```

```
    double area();  
}
```

```
public static void main (String[]  
                        args) {
```

```
    HasArea a;  
    a = new Square (5);  
    a.area();  
}
```

$$U_{ij}^1 = a_{ij}$$

$$U_{ij}^{(m+1)} = \sum_{k=1}^n U_{ik}^{(m)} + a_{kj}^-$$

$$U^4 = U^2 \times U^2$$

```

public int min(int[] a) {
    if (a.length == 1) {
        return a[0];
    }
    else {
        int min2 =
            min (Teilarray(a,
                1, a.length - 1));
        if (a[0] <= min2) {
            return a[0];
        }
        else {
            return min2;
        }
    }
}

```

$f(n) \in O(g(n)) \Leftrightarrow$

$\exists c > 0, \forall n \geq N_0, N_0 \in \mathbb{N} :$

$$f(n) \leq c \cdot g(n)$$

```
public class circle {
    private int radius;
    public circle() {
        this.radius = 1;
    }
    public circle(int radius) {
        this.radius = radius;
    }
    public double getArea() {
        return Math.PI * Math.pow(this.radius, 2);
    }
}

public interface color {
    final int black = 0;
    public int getColor();
    public void setColor(int color);
}
```



```

public class PaintedColorCircle extends Circle implements Color {
    private int color;
    public int getColor() {
        return this.color;
    }
    public void setColor(int color) {
        this.color = color;
    }
    public PaintedColorCircle() {
        super();
        this.color = black;
    }
}

public class BlackCircle extends PaintedColorCircle {
    public void setColor() {
        this.color = black;
    }
    public int getColor() {
        return black;
    }
}

public static void main (String [] args) {
    Circle c = new BlackCircle();
    System.out.println(c.getColor());
}

```

$f \in o(g(n)) \iff \forall c > 0 \exists n_0 \in \mathbb{N} : \forall n \geq n_0 : f(n) < c \cdot g(n)$

$f \in \Omega(g(n)) \iff \exists c > 0, n_0 \in \mathbb{N} : \forall n \geq n_0 : f(n) \geq c \cdot g(n)$

$f(x) = x \cdot f \in o(n^2)$

$$f(n) = \frac{n^4}{n^3 + 1} = \frac{n^4}{n^3 \cdot (1 + \frac{1}{n^3})} = \frac{n}{1 + \frac{1}{n^3}} \xrightarrow{n \rightarrow \infty} n$$

$f \in O(n)$
 $f \in \Omega(n)$ } $f \in \Theta(n)$

```
public int factorial(int n) {  
    if (n == 1)  
        return 1;  
    else {  
        return n * factorial(n-1);  
    }  
}
```