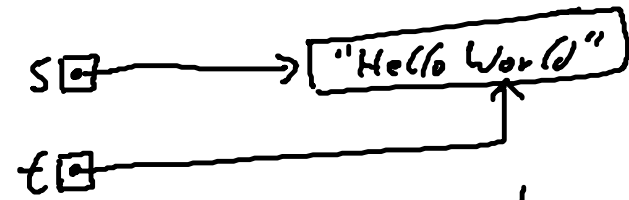


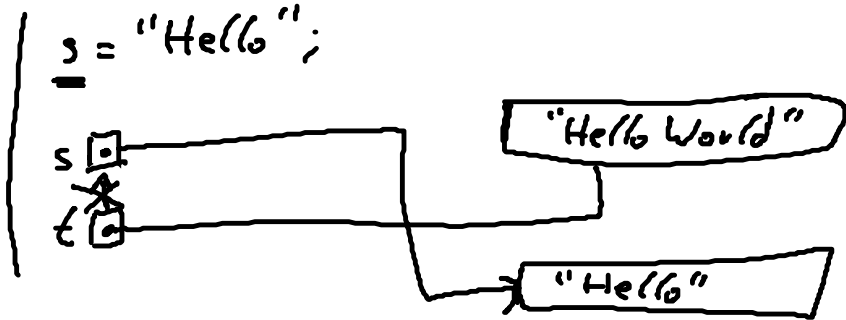
CoMa - Übung VII (05.12.12)

```
String s = "Hello World";
```

```
String t = s;
```



```
...  
s = "Hello"
```

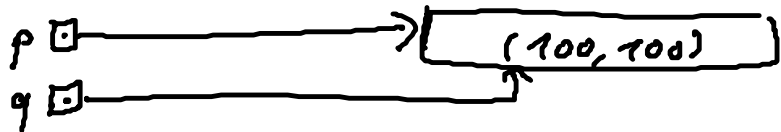


```
Point p = new Point(0,0);
```

```
Point q = p;
```



```
p.move(new Point(100,100));
```



```
(p == q) → true
```

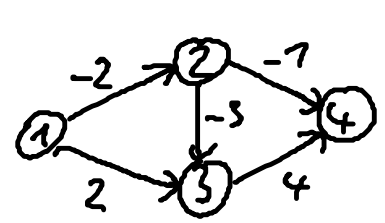
```
(p.equals(q)) → true
```

```
int i = 10;
```

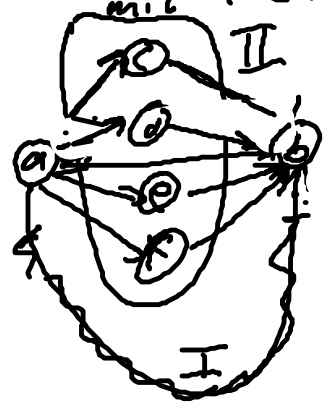
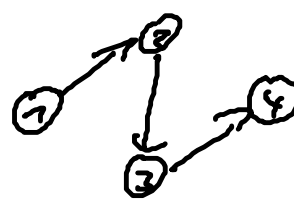
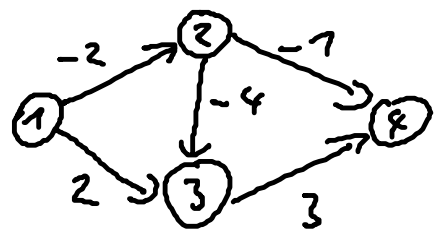
```
i 10
```

Bellman-Ford-Gleichungen:

- Dynamische Programmierung
- Kürzester Weg mit 1 Kante finden ist leicht.
- Kann aus der Distanzmatrix A abgelesen werden.
- Idee: Berechne daraus kürzeste Wege mit 1-2 Kanten (maximal 2 Kanten)



Kein negativer Zykel.
Kürzester Weg:

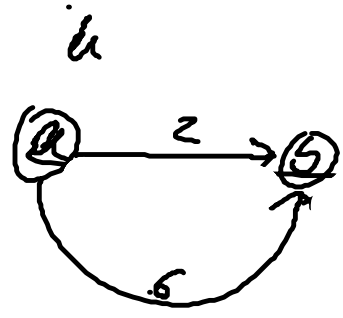


$u_{ik}^{(j)}$:= Länge eines kürzesten Weges von i nach k mit maximal j Kanten

$u_{ik}^{(1)} := a_{ik}$ (\rightarrow in Distanzmatrix $A = (a_{ij})$ nachgucken)

$a \rightarrow c \rightarrow b$ $u_{ac}^{(1)} + u_{cb}^{(1)} \rightarrow$ Länge des Weges

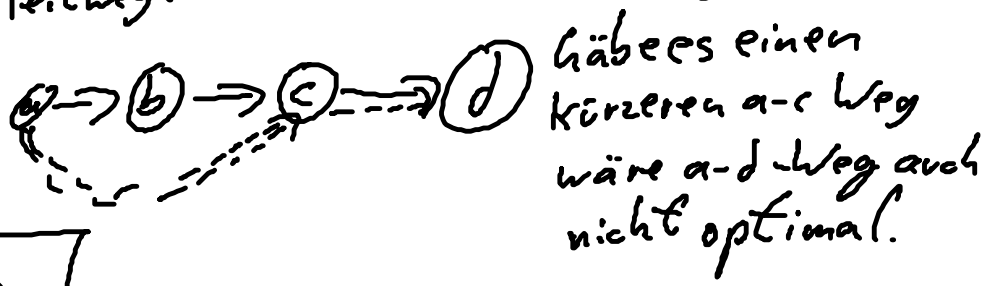
$$u_{ab}^{(2)} := \min \left\{ \underbrace{u_{ab}^{(1)}}_I, \min_{l=1, \dots, n} \left\{ \underbrace{u_{al}^{(1)}}_I + \underbrace{u_{lb}^{(1)}}_I \right\} \right\}$$



Prinzip von der optimalen Substruktur:
Teilstrukturen einer opt. Struktur sind selbst optimal.

Bsp: kürzeste Wege:

$a \rightarrow b \rightarrow c \rightarrow d \rightarrow \text{k.V. von } a \rightarrow d$
 \Rightarrow Teilwege sind auch kürzeste Wege



$$u_{ij}^{(m+1)} := \min_{k=1, \dots, n} \{ u_{ik}^{(m)} + a_{kj} \}$$

$$\{ u_{ij}^{(m)} + a_{jj} \}$$

$$a_{jj} = 0$$

$$u_{ij}^{(1)} := a_{ij}$$

$A = (a_{ij})_{i,j=1, \dots, n}$ Distanzmatrix
 $u_{ij}^{(1)} := a_{ij}$ für alle $i, j = 1, \dots, n$
 $u_{ij}^{(m+1)} := \dots$

short s = (short) 1;
 short t = (short) 3;
 short u = s + t;
 (int)

Short s = (short) 1;
 long l = s;

int a = 4;

int b = 0;

int c = a / b; → ArithmeticException → Laufzeitfehler

Point p = null; p →

p.move(...); → NullPointer Exception

int[] array = new int[4];

System.out.println(array[10]); → ArrayIndexOutOfBoundsException

new Point();

Assertion Error