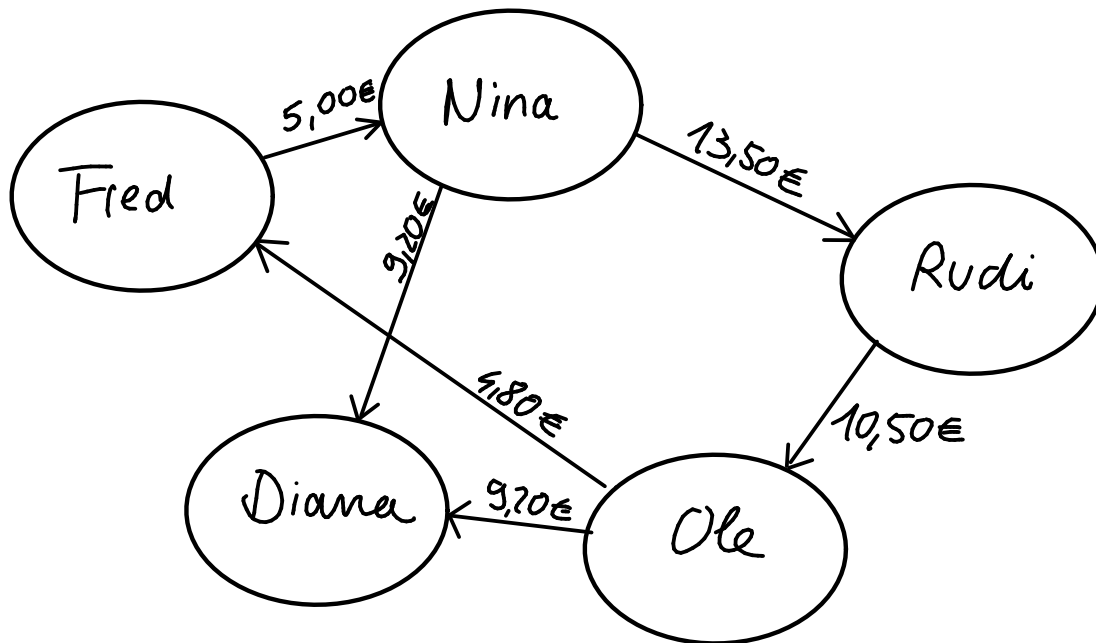


2. Schulaufgabe aus der Informatik

Musterlösung

Aufgabe 1a



Anm: Richtung nicht aus Angabe ersichtlich. (Einheitlich) umgekehrte Lösung ebenfalls i.O.

Aufgabe 1b

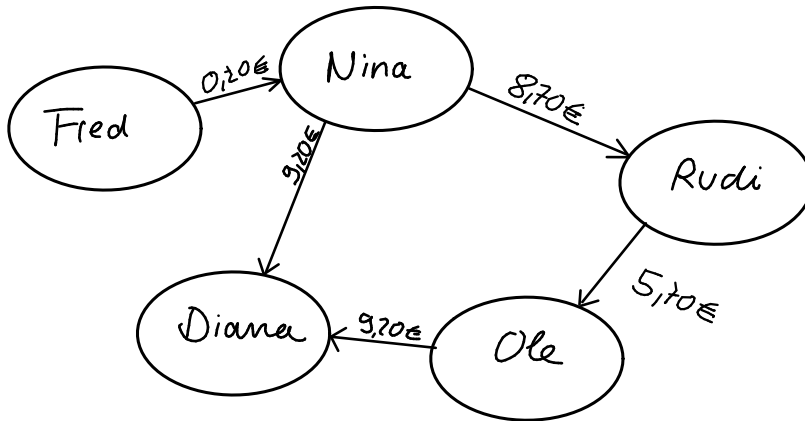
Schuldet >	F	N	O	R	D
F		5,00			
N				13,50	9,20
O	4,80				9,20
R			10,50		
D					

Anm: Auch hier entsprechend transponierte Matrix i.O.

Aufgabe 2a

Der Graph ist [✓]zyklisch, so kommt die Verklemmung zustande.

Aufgabe 2b



Aufgabe 2c

Der Graph ist [✓]gerichtet, [✓]gewichtet, zusammenhängend und unvollständig (Aum.: 2 ausreichend.)

Aufgabe 3a

```
double KontostandBerechnen ( int n ) {  
    double summeVerliehen;  
    double summeGeliehen;  
    for ( int x = 0; x ≤ schulden.length; x++ ) {  
        summeVerliehen = summeVerliehen + schulden[n][x];  
        summeGeliehen = summeGeliehen + schulden[x][n];  
    }  
    return summeVerliehen - summeGeliehen;  
}
```

Aufgabe 3b

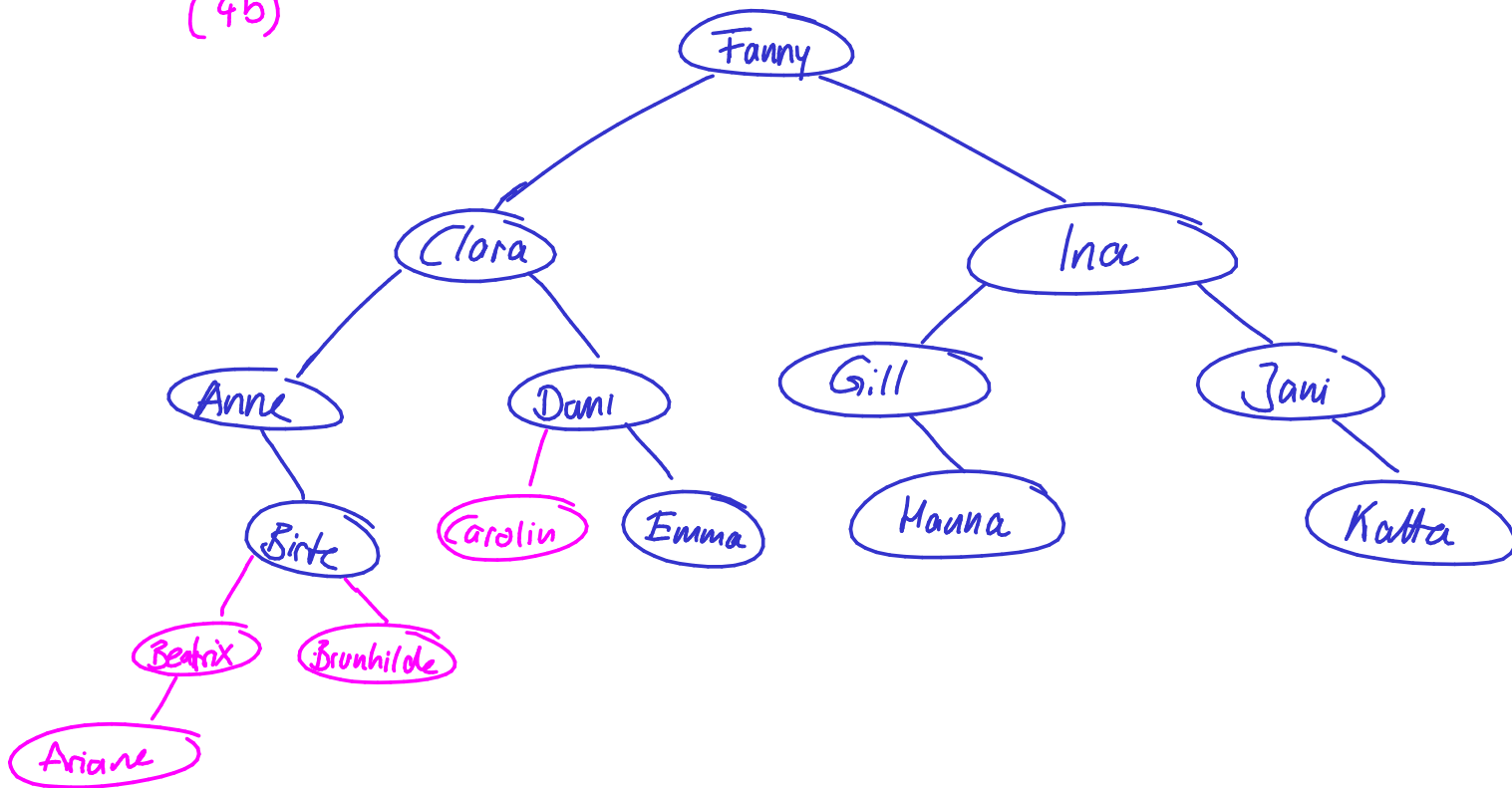
Die Aussage ist korrekt. Beispielsweise führen diese zwei Matrizen zu dem selben Kontoständen:

	A	B	C
A		1	
B			2
C	1		

	A	B	C
A		6	
B			7
C	6		

Ergo ist aus den Kontoständen alleine nicht reproduzierbar welche Schuldenverhältnisse genau vorliegen.

Aufgabe 4a (4b)



c)

$$n = 2^E - 1$$

Auf einen Baum mit 6 Ebenen passen demnach nur 63 Elemente. Ergo ist eine ~~7~~ Ebene mindestens nötig.

d) In Vereinsbaum:

```
int baumTiefeGeben ( ) {  
    if (wurzel != null) {  
        return wurzel.tiefeGeben();  
    } else {  
        return null;  
    }  
}
```

✓
✓
✓

In Mitglied:

```
int tiefeGeben() {
```

```
    int l = 0;
```

```
    if (linkerNachfolger != null) {
```

```
        l = linkerNachfolger.tiefeGeben();
```

```
    }
```

```
    int r = 0;
```

```
    if (rechterNachfolger != null) {
```

```
        r = rechterNachfolger.tiefeGeben();
```

```
    }
```

```
    if (l > r) {
```

```
        return l + 1;
```

```
    } else {
```

```
        return r + 1;
```

```
    }
```

```
}
```