

Übungsblatt 11

Aufgabe 11.3.

Hinweis: Auf Korrektheit und Vollständigkeit der Lösungen besteht keine Gewähr.

a)

- (i) Zulässig
- (ii) Nicht zulässig — Implizite Typumwandlung funktioniert nur bei primitiven Datentypen.
- (iii) Zulässig (autoboxing)
- (iv) Nicht zulässig — 1.2 ist `double`-Literal.

b)

(i) 78 - 42

$$\begin{array}{rcll} 78 & = & 0100 & 1110 \\ 42 & = & 0010 & 1010 \\ -42 & = & 1101 & 0101 + 1 \\ & = & 1101 & 0110 \end{array} \quad \begin{array}{l} \\ \\ \\ \end{array} \quad \begin{array}{rcll} & & 0100 & 1110 \\ & & + & 1101 & 0110 \\ & & ===== \\ (1) & & 0010 & 0100 & = 36 \end{array}$$

(ii) -12 + 50

$$\begin{array}{rcll} 12 & = & 0000 & 1100 \\ -12 & = & 1111 & 0011 + 1 \\ & = & 1111 & 0100 \\ 50 & = & 0011 & 0010 \end{array} \quad \begin{array}{l} \\ \\ \\ \end{array} \quad \begin{array}{rcll} & & 1111 & 0100 \\ & & + & 0011 & 0010 \\ & & ===== \\ (1) & & 0010 & 0110 & = 38 \end{array}$$

c)

```
boolean isSymmetric(final double[][] A) {
    final int n = A.length;
    for (int i = 1; i < n; i++) {
        for (int j = 0; j < i; j++) {
            if (A[i][j] != A[j][i]) {
                return false;
            }
        }
    }
    return true;
}
```

d)

(i)

```
boolean isSorted(Node node) {
    if ((node == null) || (node.next == null)) {
        return true;
    }
    return (
        (node.value >= node.next.value) &&
        isSorted(node.next)
    );
}
```

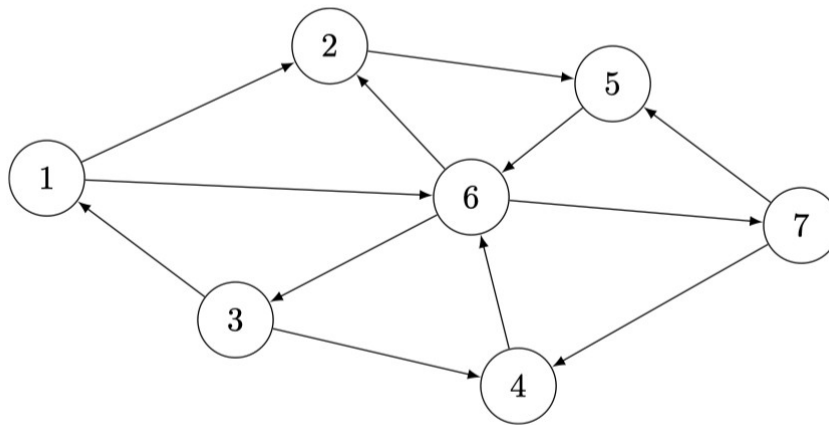
(ii)

```
Node joinPairwise(Node node) {
    if (node == null) {
        return null;
    }
    if (node.next == null) {
        return new Node(node.value, null);
    }
    return new Node(
        node.value + node.next.value,
        joinPairwise(node.next.next)
    );
}
```

e)

(i) Gerichtet, die Adjazenzmatrizen ungerichteter Graphen sind symmetrisch, diese ist es nicht.

(ii)



f)

(i)

```
public Pair(T first, U second) {  
    this.first = first;  
    this.second = second;  
}
```

(ii)

```
Pair<String, Integer> p = new Pair<>();
```

(iii)

```
public interface PairConverter<T, U, V, W> {  
    public Pair<V, W> convert(Pair<T, U> pair);  
}
```

oder

```
public interface PairConverter {  
    public <T, U, V, W> Pair<V, W> convert(Pair<T, U> pair);  
}
```

g)

$$O(n) \cdot O(n) \cdot O(\log n) = O(n^2 \log n)$$