.

Aufgabe 5.1

Im Material zu diesem Aufgabenblatt ist die Klasse `TimeAndAgain.java` enthalten. Sie enthält eine Methode

```
static boolean f(boolean[] b).
```

Was ist die Laufzeit dieser Methode im schlechtesten Fall (worst case) als Funktion der Länge n des Arrays `b` (die genaue Funktion, keine O-Notation)? Erläutern Sie kurz, wie Sie auf diese Laufzeit gekommen sind. (3 Punkte)

Aufgabe 5.2

Für welche der folgenden Paare von Funktionen f und g gilt $f \in O(g)$?

(a) $f(n) = 3n^3 + 200$, $g(n) = \frac{1}{2}n^3 - 10$

(b) $f(n) = \sqrt{n} - 6$, $g(n) = 20\sqrt[4]{n} + 36$

(c) $f(n) = 2n^2 - 8n + 2$, $g(n) = n^3 - 17n^2 - 23451$

(3 Punkte)

Aufgabe 5.3

Schreiben Sie eine Klasse `StraightLine.java`. Diese Klasse soll zwei globale Variablen `slope` und `intercept` vom Typ `double` haben. Diese beschreiben den Anstieg und den y-Abschnitt der Geraden. Beide Variablen sollen durch den Zugriffsregler `private` vor Zugriffen von außerhalb der Klasse geschützt werden. Die Klasse soll ferner über einen Konstruktor der Form

```
public StraightLine(double slope, double intercept)
```

verfügen, der den Anstieg und den y-Abschnitt der Geraden initialisiert. Außerdem soll es in der Klasse die folgenden Methoden geben:

- `public double getSlope()`, welche den Anstieg der Geraden als Rückgabewert hat.
- `public double getIntercept()`, welche den y-Abschnitt der Geraden als Rückgabewert hat.
- `public void setSlope(double slope)`, welche der globalen Variablen `slope` den übergebenen Wert zuweist.
- `public void setIntercept(double intercept)`, welche der globalen Variablen `intercept` den übergebenen Wert zuweist.
- `public double valueAt(double x)`, welche den Funktionswert der linearen Funktion $g(x) =$ `slope` $\cdot\, x +$ `intercept` an der Stelle `x` zurückgibt.

(7 Punkte)

Rückseite beachten!

Zusatzaufgabe 5.4

Wir schauen uns wohlgeformte Klammerausdrücke an, die zwei verschiedene Arten von Klammern enthalten, nämlich eckige und runde. Ein Beispiel für einen solchen Ausdruck ist `()[()(())][[]]`. Wie könnte eine induktive Definition für diese Art von Klammerausdrücken aussehen? Wie könnte man das Programm `MatchBracket` aus der Vorlesung so erweitern, dass es bei Eingabe eines beliebigen Strings, der nur eckige und runde Klammern enthält, entscheidet, ob es sich um einen wohlgeformten Ausdruck handelt oder nicht? (1+Punkt)