

4. ÜBUNG ZUR VORLESUNG PROGRAMMIERUNG II, SS 2003

Aufgabe 1: $\mathcal{O}$ -Notation, Teil 1 (10 Punkte)

Beweisen oder widerlegen Sie die folgenden Aussagen:

- $f(x) \in \mathcal{O}(g(x)) \Leftrightarrow g(x) \in \Omega(f(x))$
- $o(f(x)) \cap \omega(f(x)) = \emptyset$
- $f(x) + g(x) \in \Theta(\min(f(x), g(x)))$
- $f(x) \in \Theta(f(\frac{x}{2}))$
- Für $g(x) \in o(f(x))$: $f(x) + g(x) \in \Theta(f(x))$

Hinweis: Sie können hierbei die folgenden Beziehungen verwenden:

- $\lim_{x \rightarrow \infty} \frac{|f(x)|}{|g(x)|} = C$ mit $0 \leq C < \infty \Rightarrow f(x) \in \mathcal{O}(g(x))$
- $\lim_{x \rightarrow \infty} \frac{|f(x)|}{|g(x)|} = 0 \Rightarrow f(x) \in o(g(x))$

Aufgabe 2: $\mathcal{O}$ -Notation, Teil 2 (10 Punkte)

Beweisen Sie die folgenden Behauptungen:

- $x^4 \in o(x^5)$
- $x^4 + 42x^3 \in \Theta(x^4)$
- $2x^6 \in \mathcal{O}(x^6) \setminus o(x^6)$
- $\forall a \in \mathbb{N} : e^x \in \omega(x^a)$
- $\ln(x) \in o(x)$

Achten Sie bei Ihren Beweisen besonders auf formale Korrektheit.

Aufgabe 3: $\mathcal{O}$ -Notation, Teil 3 (10 Punkte)

Ordnen Sie die folgenden Funktionen aufsteigend nach Ihrer Wachstumsgeschwindigkeit, d.h. finden Sie eine Anordnung $f_1, f_2, \dots, f_5$, so dass für $i \in 1, \dots, 4$: $f_i \in \mathcal{O}(f_{i+1})$.

$$e^x, \ln(\ln(x)), x!, \sqrt{x}, x^x$$

Begründen Sie Ihre Aussagen!

Aufgabe 4: Laufzeitberechnung (10 Punkte)

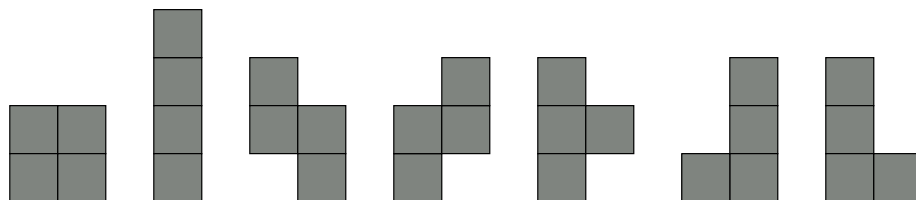
Der folgende C++-Code implementiert InsertionSort für einen `vector<int> v`:

```
for (j = 1; j < v.size(); j++)
{
    int key = v[j];
    int i = j - 1;
    while ((i >= 0) && (v[i] > key))
    {
        v[i+1] = v[i];
        i--;
    }
    v[i+1] = key;
}
```

Zeigen Sie: für die *worst-case Laufzeit* $T(n)$ von InsertionSort gilt $T(n) \in \theta(n^2)$, wobei $n = v.size()$ die Länge des Vektors, also die Problemgröße bezeichnet.

Zusatzaufgabe 5: Klassenhierarchien (10 Bonuspunkte)

Das Computerspiel *Tetris* besteht aus einem *Spielfeld*, welches in 20 Zeilen und 10 Spalten eingeteilt ist, sowie aus verschiedenen geformten *Steinen* (siehe Bild). In jeder *Runde* fällt dabei ein zufällig gewählter Stein solange nach unten, bis er entweder gegen einen schon gefallenem Stein oder den Boden des Feldes stößt. Dabei kann der Stein in jedem Schritt im oder gegen den Uhrzeigersinn gedreht werden, fällt dann eine Zeile nach unten und wird neu *gezeichnet* (das Löschen von vollen Zeilen brauchen Sie hier nicht zu betrachten).



Entwerfen Sie eine Klassenhierarchie für das Spiel Tetris. Bitte beachten Sie, dass Sie Ihre Klassenhierarchie **nicht** implementieren müssen! Stellen Sie sie statt dessen in Form eines Klassendiagramms, wie es Ihnen aus der Vorlesung bekannt sein sollte, graphisch dar.

Abgabe: 23.05.2003