

Algorithmen & Datenstrukturen

Blatt 4

Dr. Matthias Thimm

Tina Walber, Leon Kastler, Martin Leinberger und Maximilian Strauch

Fachbereich Informatik, Universität Koblenz-Landau

7. Dezember 2013

1 Experimentelle Bestimmung von Aufwandsklassen (9 Punkte)

Im Folgenden sollen Sie experimentell die Aufwandsklasse des InsertionSort-Algorithmus bestimmen. Gehen Sie dafür wie folgt vor:

- a) **Implementieren von InsertionSort (3 Punkte)** Adaptieren Sie den InsertionSort-Algorithmus aus der Vorlesung (Foliensatz 06, Folien 10ff), so dass er mit beliebigen Listen¹ statt Arrays umgehen kann. Benutzen Sie für die Sortierfunktion die folgende Signatur:
`public static void sort(List<Integer> list)`
- b) **Messung der Geschwindigkeit (3 Punkte)** Messen Sie die Geschwindigkeit Ihrer Sortierfunktion für Listen mit $n=100$, $n=200$, ..., $n=2000$ zufällig sortierten Elementen. Ermitteln Sie für jede Listengröße aus mehreren Messungen die durchschnittlich benötigte Zeit (in Millisekunden) zum Sortieren. Geben Sie den Sourcecode ihres Messframeworks ab. Geben Sie außerdem das Ergebnis zweier unterschiedlicher Messreihen ab: Eine Messreihe mit der Geschwindigkeit, wenn Sie der Sortierfunktion eine ArrayList² übergeben. Eine zweite Messreihe mit der Geschwindigkeit für LinkedList³.
- c) **Aufwandsklasse bestimmen (3 Punkte)** Verwenden Sie Ihre Messreihen, um die Aufwandsklasse Ihrer Implementierung von InsertionSort zu bestimmen. Zeigen Sie dafür anhand Ihrer Messreihe, dass es ein c und n_0 gibt, so dass $\forall n \geq n_0 : g(n) \leq c \cdot f(n)$ (Foliensatz 04, siehe Folie 14). Dabei ist $g(n)$ die von Ihnen gemessene Zeit für eine Liste der Größe n und $f(n)$ die von Ihnen vermutete Aufwandsklasse. Vergleichen und diskutieren Sie die Ergebnisse für die beiden Messreihen.

¹<http://docs.oracle.com/javase/6/docs/api/java/util/List.html>

²<http://docs.oracle.com/javase/6/docs/api/java/util/ArrayList.html>

³<http://docs.oracle.com/javase/6/docs/api/java/util/LinkedList.html>



2 Laufzeitanalyse: Code geben und Laufzeit berechnen (3 Punkte)

Bestimmen Sie für die folgenden Methoden alg1, alg2 und alg3 jeweils:

- ob sie terminieren für beliebige Eingaben
- wenn sie terminieren, die Aufwandklasse.

Alle Eingaben der Algorithmen kommen immer aus dem Bereich der natürlichen Zahlen. Um zu zeigen, dass ein Algorithmus nicht terminiert genügt die Angabe eines einfachen Beispiels.

```
a) void alg1(int n) {
    int m = 1;
    int num = n*n;

    while (num != 0) {
        num = num - m;
        m = (m+1)*m;
    }
}

b) void alg2(int n) {
    int m = 1;
    int i = n*n;
    int j;
    int p = 0;

    while (i > 1) {
        j = 1;
        while (0 <= n-j) {
            p = 2*j;
            j = j + 1;
        }
        i = i / 2;
    }
}
```



```
c) void alg3(int n) {
    int i = n;
    int k = n;
    int j = 0;

    for (j = 0; j < n; j++) {
        if (j mod 2 == 0)
            k = k / 2;
        else
            k = k - 1;

        while (i > 0) {
            int p = 1;
            while (p < n*n) {
                k = k + (k * i) mod p;
                p = p + 1;
            }
            i = i - 1;
        }
    }
}
```



3 Wachstumsverhältnisse zwischen Funktionen (5 Punkte)

Fülle in der folgenden Tabelle von Funktionen die Wachstumsverhältnisse zwischen Funktionen aus.

Die Tabelle ist so zu verstehen: wenn die Funktion $f(n)$ in der Zeile y und die Funktion $g(n)$ in Spalte x stehen, so tragen Sie alle Zeichen α aus $\{O, \Omega, \Theta\}$ in die Zelle (x, y) ein, so dass $f(n) \in \alpha(g(n))$ gilt. Also z.B. $13n \in O(\log(n))$. Füllen Sie alle Felder der Tabelle aus. (1 Punkt pro Zeile, pro Fehler 0.5 Abzug, mind. 0 Punkte pro Zeile)

Funktion	1000	$\frac{n}{2}$	$\log(n)$	\sqrt{n}	$\frac{4^n}{4}$	$5n^2$
$13n$						
$\frac{\log(n)}{2}$						
$3n^2 + 5$						
e^n						
$(2 - (-1)^n)n$						



4 Vergleich von Aufwandsklassen (3 Punkte)

Wählen Sie für a) $13n \in \alpha(\frac{n}{2})$, b) $3n^2 + 5 \in \alpha(\frac{4^n}{4})$ und c) $e^n \in \alpha(\log(n))$ je ein α aus $\{O, \Omega, \Theta\}$ (siehe vorhergehende Aufgabe). Zeigen Sie die Richtigkeit Ihrer Aussage. (Tip: Sie können ein geltendes n_0 und c angeben oder $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)}$ bzw. $\lim_{n \rightarrow \infty} \frac{g(n)}{f(n)}$ berechnen) (1 Punkt pro Beweis)

