Prof. Dr. Gerd Stumme, Folke Eisterlehner, Dominik Benz
Fachgebiet Wissensverarbeitung

28./30.04.2009

2. Präsenzübung „Algorithmen und Datenstrukturen“

Sommersemester 2009

1 Berechenbarkeit

Diskutieren Sie zu jeder der folgenden Problemstellungen, ob Sie sich algorithmisch lösen lässt.

Ja	Nein	
☐	☐	Gegeben ein beliebiges Java-Programm `P` und beliebige Eingabe `x`. Hält `P` auf Eingabe `x`?
☐	☐	Gegeben ein beliebiges Java-Programm `P`. Hält `P` auf Eingabe `0`?
☐	☐	Gegeben ein beliebiges Java-Programm `P` und beliebige Eingabe `x`. Hält `P` auf Eingabe `x` nach weniger als 10^x Schritten?
☐	☐	Gegeben zwei beliebige Java-Programme `P` und `Q`. Berechnen `P` und `Q` dieselbe Funktion?
☐	☐	Gegeben zwei beliebige Java-Programme `P` und `Q`. Erzeugen `P` und `Q` auf Eingabe `42` dieselbe Ausgabe?

2 Komplexität

Achten Sie bei allen Teilaufgaben auf eine korrekte Notation und verwenden Sie möglichst einfache Laufzeitformeln, z.B. $O(n^2)$ statt $O(3n^2 + 7n - 5)$.

a) Gegeben sind folgende Funktionen $f_i \colon \mathbb{N} \to \mathbb{N}$, $i = 1, \ldots, 5$:

$$f_1(n) := n \qquad f_2(n) := \tfrac{1}{3}n^3 + 10n^2 - n \qquad f_3(n) := n \log n \qquad f_4(n) := n^2 \qquad f_5(n) := e^n$$

Untersuchen Sie, ob $f_i \in O(f_{i+1})$ für $i = 1, \ldots, 4$ gilt. Begründen Sie Ihre Behauptung. Dabei können Sie folgende Eigenschaften verwenden:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty \implies f \in O(g)$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \implies f \notin O(g)$$

b) Welche worst-case Laufzeit in O-Notation hat der folgende Algorithmus zum Sortieren eines `int`-Arrays `a`? Begründen Sie.

```
int n = a.length;
int[] count = new int[n];
int[] result = new int[n];
for (int i = 0; i < n; ++i) { count[i] = 0; }
for (int i = 0; i < n - 1; ++i) {
```

```
    for (int j = i + 1; j < n; ++j) {
      if (a[i] < a[j]) {
        ++count[j];
      } else {
        ++count[i];
      }
    }
  }
  for (int i = 0; i < n; ++i) {
    result[count[i]] = a[i];
  }
```

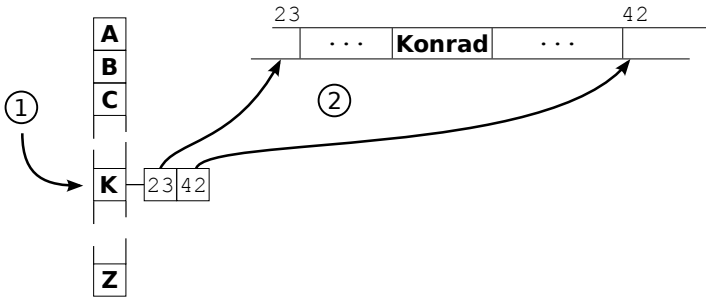
c) Welche worst-case Laufzeit in O-Notation hat der folgende Algorithmus für den Primzahltest? Begründen Sie.

```
boolean isPrime(int n) {
  if (n > 2 && n % 2 == 0) { return false; }
  for (int i = 3; i * i <= n; i += 2) {
    if (n % i == 0) { return false; }
  }
  return true;
}
```

d) Diskutieren Sie die Laufzeit des folgenden Suchverfahrens für Zeichenketten in sortieren Feldern:

In einem zusätzlichen Index-Feld steht zu jedem Buchstaben im Alphabet Start– und Endadresse des Bereichs im sortierten Feld, der alle Schlüssel mit dem entsprechenden Anfangsbuchstaben enthält.

Wird nun nach einer Zeichenkette $a_1 \ldots a_n$ gesucht ($a_1, \ldots, a_n$ sind Buchstaben im Alphabet), wird (1) zunächst im Index-Feld der Bereich bestimmt, in dem sich alle Schlüssel befinden, die mit dem Buchstaben a_1 beginnen. Dort wird dann (2) mit *binärer Suche* fortgesetzt.



Index-Suche nach dem Schlüssel „Konrad“