Prof. Dr. V. Linnemann
Stefan Werner
Universität zu Lübeck
Institut für Informationssysteme

Lübeck, den 2. Mai 2013

Algorithmen und Datenstrukturen

Sommersemester 2013

5. Übungsblatt

Hinweis: Aufgrund des Feiertages sind die Lösungen zu diesem Übungsblatt bereits am Mittwoch, den 8. Mai 2013 bis 10 Uhr abzugeben.

Aufgabe 1: Divide and Conquer

Seien x und y zwei n-Bit Integer, welche multipliziert werden sollen. Eine Methode $mult(x, y)$, um die beiden Zahlen unter Verwendung des Teile-und-Herrsche-Prinzips zu multiplizieren, geht wie folgt vor:

Die beiden Binärzahlen werden in eine linke und eine rechte Hälfte geteilt, so dass gilt:
$x = 2^{n/2} * x_L + x_R$
$y = 2^{n/2} * y_L + y_R$

Damit folgt für das Produkt der beiden Zahlen:
$xy = 2^n x_L y_L + 2^{n/2}(x_L y_R + x_R y_L) + x_R y_R$

So reduziert sich das Problem auf die Multiplikation kleinerer Zahlen, da x_L, x_R, y_L, y_R nur noch jeweils $n/2$ Bitstellen besitzen. Dieses Vorgehen wird rekursiv fortgesetzt bis jeder Teil einstellig ist und die Multiplikation einfach durchgeführt werden kann. Die Multiplikation mit einer Zweierpotenz ergibt sich durch einen einfachen Links-Shift und das Hinzufügen entsprechend vieler Nullen am Ende.

a) Berechnen Sie nachvollziehbar $11_2 * 10_2$ nach dieser Methode. (4 Punkte)

b) Zeichnen Sie den Aufrufbaum des Aufrufs $mult(1101_2, 0110_2)$. (4 Punkte)

Aufgabe 2: Java - Dynamisches Programmieren

Die folgende Funktion $P(n, k)$ berechnet die Binomialkoeffizienten für $0 \leq k \leq n$:

$$P(n,k) = \begin{cases} 1, \text{ falls } k = 0 \text{ oder } k = n \\ P(n-1, k-1) + P(n-1, k), \text{ sonst} \end{cases}$$

$P(n,k)$ ist die k-te Zahl in der n-ten Zeile im Pascalschen Dreieck:

```
                              | n:
             1                | 0
           1   1              | 1
         1   2   1            | 2
       1   3   3   1          | 3
     1   4   6   4   1        | 4
   1   5  10   10  5   1      | 5
-----------------------------
k: 0   1   2    3  4   5
```

Eine naive rekursive Implementierung sieht wie folgt aus:

```
public static int binomialRek(int n, int k) {
    if (n < 0 || k < 0 || n < k) return 0;
    if (k == 0 || k == n) return 1;
    return binomialRek(n-1, k-1)+binomialRek(n-1, k);
}
```

Implementieren Sie eine Java-Methode `public static int binomialDyn(int n, int k)`, die $P(n,k)$ nach dem Prinzip des dynamischen Programmierens berechnet. Beschreiben Sie ausführlich Ihre Methode!
Schätzen Sie die Laufzeit Ihrer Methode in O-Notation ab. (9 Punkte)

Aufgabe 3: Suchen und selbstanordnende Listen

Gegeben sei die Zahlenfolge $A = [7, 42, 8, 110, 6, 3, 66]$ und die Zugriffsfolgen
$Z_1 = [110, 3, 6, 42, 66, 7, 8]$ und $Z_2 = [6, 3, 66, 6, 3, 66, 6, 3, 66]$.

a) Geben Sie jeweils jeden Schritt und die Anzahl der Vergleiche für die Suche nach 42 und nach 23 in A bei Verwendung der *linearen Suche* und der *binären Suche* an. (2 Punkte)

Geben Sie die Anzahl der Vergleiche sowie die aktualisierten Inhalt der selbstanordnenden Liste bei jedem Zugriff für die Zugriffsfolgen Z_1 und Z_2 bei Verwendung der

b) MF-Regel (Move-to-Front) und (3 Punkte)

c) FC-Regel (Frequency-Count) an. (3 Punkte)

Beachten Sie die Hinweise auf dem ersten Übungsblatt.

Abgabetermin: Mittwoch, den 8. Mai bis 10 Uhr im Institut für Informationssysteme, 2. OG im Informatik-Neubau (Raum 2.071, Küche neben Sekretariat)