



Algorithmen und Datenstrukturen 1
186.089 VO 3.0 + 186.114 UE 2.0
Vorlesungsprüfung / 3. Übungstest SS 2006
30. Juni 2006

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: Vorname:

Matrikelnummer: Studienkennzahl:

Anzahl abgegebener Zusatzblätter:

Prüfung soll gewertet werden als (Mehrfachnennungen möglich):

- Vorlesungsprüfung
- 3. Übungstest
- Vorlesungsprüfung **und** 3. Übungstest (**Default**)

Legen Sie während des Tests Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten können. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

Die Verwendung von Taschenrechner, Mobiltelefonen, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

	A1:	A2:	A3:	A4:	A5:	Summe:
Erreichbare Punkte:	10	10	10	10	10	50
Erreichte Punkte:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Viel Erfolg!

Aufgabe 1.A: $\Omega/O/\Theta$ -Notation

(10 Punkte)

a) (4 Punkte)

Beweisen oder widerlegen Sie folgende Aussagen:

- $f(n) = O(g(n)) \Rightarrow \log f(n) = O(\log g(n))$
- $f(n) = O(h(n)) \wedge g(n) = O(h(n)) \Rightarrow h(n) = \Omega(f(n) + g(n))$

b) (2 Punkte)

Geben Sie an, ob $f(n) = O(g(n))$ oder $g(n) = O(f(n))$ oder beides gilt. Beweisen Sie Ihre Aussage mittels der Definition der O-Notation und geben Sie entsprechende Konstanten an.

- $f(n) = n \cdot (n + 5) + 4 \cdot n^3 \quad g(n) = n((n - 1)(n - 15))$

c) (4 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} \frac{5n}{(n^3 - 3n)} + n^n \cdot 3 \cdot n^2, & n \geq \sqrt[3]{10000} \\ \frac{13 \cdot n}{n^3} + \frac{6n}{(n - 2)} + n^3, & n < \sqrt[3]{10000} \end{cases}$$

Kreuzen Sie in der folgenden Tabelle die zutreffenden Felder an:

$f(n)$ ist	$\Theta(\cdot)$	$O(\cdot)$	$\Omega(\cdot)$	keines
n^n				
n^{2n}				
$n!$				

(Anmerkung: Jede Zeile wird jeweils nur dann gewertet, wenn alle Felder der Zeile richtig ausgefüllt sind.)

Aufgabe 2.A: Bäume/Algorithmus von Fitch

(10 Punkte)

Es soll der aus dem Bereich der Bioinformatik kommende Algorithmus von Fitch implementiert werden. Dieser arbeitet auf einem so genannten *Abstammungsbaum*, bei dem es sich um einen binären Baum handelt, wo jeder Knoten entweder ein Blatt (Knoten ohne Nachfolger) oder ein innerer Knoten mit immer genau zwei Nachfolgern ist. Sei W ein Verweis auf den Wurzelknoten.

Jeder Knoten K besitzt die Verweise $K.left$ und $K.right$ auf die eventuell vorhandenen Nachfolger, sowie eine Menge von Beschriftungen $K.B$ aus dem Alphabet $\Sigma = \{A, C, G, T\}$. Anfangs sind nur die Blätter mit jeweils einer einzelnen Beschriftung versehen. Die inneren Knoten repräsentieren hypothetische Vorfahren, deren Beschriftungen der Algorithmus von Fitch bestimmt.

Der Algorithmus von Fitch läuft in zwei Phasen ab:

Schritt 1: Aufbau potenzieller Beschriftungen für alle inneren Knoten K mit Kindern X und Y , beginnend bei den bereits beschrifteten Blättern:

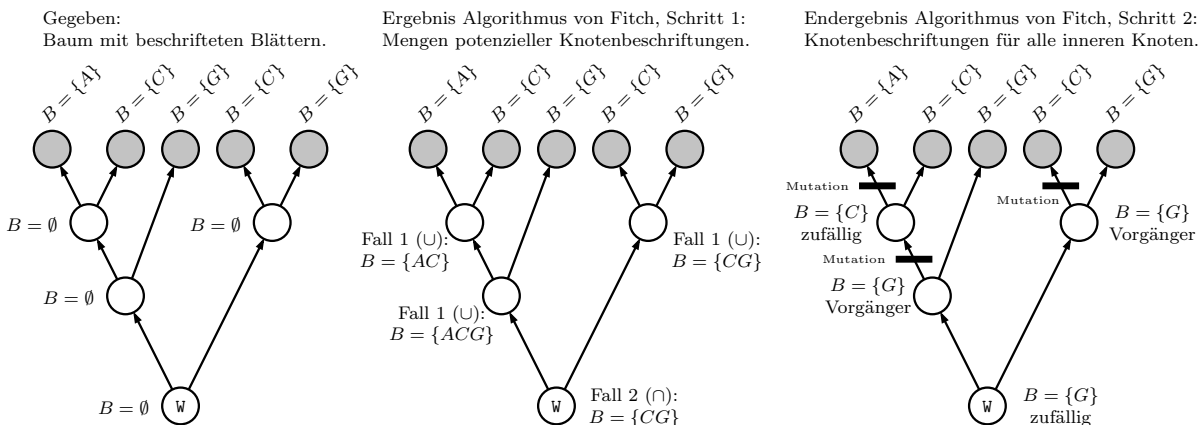
Fall 1: Durchschnitt der Beschriftungen von X u. Y leer:

$$X.B \cap Y.B = \emptyset \rightarrow K.B = X.B \cup Y.B$$

Fall 2: Durchschnitt nicht leer: $X.B \cap Y.B \neq \emptyset \rightarrow K.B = X.B \cap Y.B$

Schritt 2: Endgültige Beschriftung der inneren Knoten K , beginnend bei der Wurzel:

Sei b die Beschriftung des Vorgängerknotens von K . Falls b Element der Menge $K.B$ ist, dann wird $K.B$ auf diese Beschriftung reduziert, d.h. $K.B = \{b\}$. Andernfalls wird eine beliebige Beschriftung z aus $K.B$ gewählt, d.h. $K.B = \{z\}$, $z \in K.B$, z zufällig gewählt. Die Beschriftung der Wurzel W (hat keinen Vorgänger) wird zufällig aus $W.B$ genommen.



Schreiben Sie nun detaillierten Pseudocode für folgende *rekursive* Prozeduren und geben Sie jeweils auch den initialen Aufruf zum Starten der Rekursion mit allen konkreten Parametern an. Sie können davon ausgehen, dass Mengenoperationen (wie \cup oder \cap) definiert sind.

- (5 Punkte) Die *rekursive* Prozedur `Fitch1(...)`, die den ersten Schritt des Algorithmus von Fitch implementiert. Konkret: Traversieren Sie den gegebenen Baum in geeigneter Reihenfolge und bauen Sie in allen inneren Knoten K die zugehörigen Mengen $K.B$ der potenziellen Beschriftungen auf.
- (5 Punkte) Die *rekursive* Funktion `Fitch2(...)`, die den zweiten Schritt des Algorithmus von Fitch implementiert und die Anzahl der Mutationen (Kindknoten hat eine andere Beschriftung als Elternknoten) zurückliefert. Konkret: Gehen Sie davon aus, dass Schritt 1 des Algorithmus bereits erfolgreich abgeschlossen wurde. Traversieren Sie den Baum nun in geeigneter Reihenfolge und weisen Sie allen inneren Knoten K genau eine Beschriftung aus der zugehörigen Menge $K.B$ – dem Algorithmus von Fitch folgend – zu.

Aufgabe 3.A: Sortieren**(10 Punkte)**

a) (5 Punkte)

Sortieren Sie die in der Tabelle angeführte Zahlenfolge **aufsteigend** mittels Quicksort. Verwenden Sie dabei als Pivotelement jeweils das **erste Element** einer Teilfolge. Schreiben Sie die bearbeiteten Teilfolgen und die zugehörigen Pivotelemente nach jedem kompletten Aufteilungsschritt in die Tabelle und markieren Sie die Pivotelemente deutlich.

Hinweis: Schreiben Sie Teilfolgen gleicher Rekursionstiefe jeweils in dieselbe Zeile (für die Lösung sind nicht notwendigerweise alle Zeilen der Tabelle auszufüllen).

9	22	3	30	27	12	33	47	19	21

b) (2 Punkte)

- (b.1) (1 Punkt) Geben Sie eine Folge von $n = 5$ Zahlen an, die den Worst-Case für **absteigend** sortierendes Quicksort mit Pivotelement am **Ende** einer Teilfolge repräsentiert. Wie lautet die Worst-Case Abschätzung von Quicksort bezüglich der Laufzeit in Θ -Notation in Abhängigkeit von n ?
- (b.2) (1 Punkt) Verändert sich der Worst-Case bzw. die Eingabefolge für den Worst-Case aus (b.1), wenn man als Pivotelement das **erste Element** einer Teilfolge verwendet? Begründen Sie Ihre Antwort.

c) (2 Punkte)

Sie haben unterschiedliche Sortierverfahren mit Worst-Case Laufzeiten von $O(n)$ bis $O(n^2)$ kennengelernt. Nennen Sie zwei Voraussetzungen linearer Verfahren, die den allgemeinen Einsatz einschränken.

Aufgabe 4.A: Hashing**(10 Punkte)**

a) (7 Punkte)

In der Vorlesung wurde im Zusammenhang mit Double Hashing die Verbesserung nach Brent – ein sehr robustes und schnelles Verfahren für den Umgang mit Kollisionen – vorgestellt. Im Folgenden sei gegeben:

$$h_1(k) = k \bmod 7$$

$$h_2(k) = 1 + (k \bmod 5)$$

Zeigen Sie die Entwicklung der anfangs leeren Hashtabelle H mit einer Länge von $m = 7$, wenn sukzessive die Schlüssel

$$\langle 16, 19, 22, 8, 13, 9, 18 \rangle$$

in genau dieser Reihenfolge eingefügt werden.

b) (3 Punkte)

Wir betrachten Double Hashing mit der Divisions-Rest-Methode auf einer Tabelle der Größe p (p eine Primzahl). Entscheiden Sie, welche der folgenden Vorschläge als zweite Hashfunktion für das Sondieren bei Double Hashing brauchbar ist, wenn die allgemeine Hashfunktion folgende Form hat:

$$h(k, i) = (h_1(k) + i \cdot h_2(k)) \bmod m \text{ oder}$$

$$h(k, i) = (h_1(k) + s(k, i)) \bmod m.$$

Begründen Sie Ihre Antwort.

- $h_2(k) = k \cdot p \bmod p$
- $s(i, k) = ((k + i) \bmod p) + 1$
- $s(i, k) = ((k \cdot i) \bmod q) + 1$ sei q die größte Primzahl kleiner als p

Aufgabe 5.A: Verbesserungsheuristiken

(10 Punkte)

- (3 Punkte)
Beschreiben Sie in einem kurzen Satz, was man unter der Lokalität einer Nachbarschaft versteht. Warum ist diese Eigenschaft wichtig für eine lokale Suche?
- (4 Punkte)
Seien $f(x_1)$ bis $f(x_n)$ die Fitnesswerte der n Kandidatenlösungen x_1, \dots, x_n in der aktuellen Population eines evolutionären Algorithmus. Beschreiben Sie in kurzen Worten die Funktionsweise der *fitnessproportionalen Selektion*. Wie lautet die Selektionswahrscheinlichkeit $p_s(x_i)$ für eine Kandidatenlösung x_i , $i = 1, \dots, n$? Gehen Sie von einem Maximierungsproblem aus; weiters gilt $\forall i : f(x_i) > 0$.
- (3 Punkte)
Beschreiben Sie in wenigen Sätzen das Prinzip von Simulated Annealing. Was ist insbesondere der wesentliche Unterschied zur einfachen lokalen Suche?