Klausur Berechenbarkeit und Algorithmik SS 2014

Iwanowski 18.08.2014

<u>Hinweise:</u>

Bearbeitungszeit: 180 Minuten

Erlaubte Hilfsmittel: keine

Bitte notieren Sie Ihre Antworten ausschließlich auf dem Aufgabenblatt. Bei Bedarf benutzen Sie die Rückseite. Für Skizzen und Entwürfe steht ebenfalls die Rückseite zur Verfügung. Entwürfe, die nicht gewertet werden sollen, sind durchzustreichen.

Insgesamt gibt es 72 Bewertungseinheiten (BE) zu erzielen. Zum Bestehen benötigen Sie mindestens 36 BE.

Hinweis: Die Verteilung der Bewertungseinheiten entspricht ungefähr der Zeit, die Sie für die Lösung der jeweiligen Aufgabe aufwenden sollten.

Viel Erfolg!

Aufgabe 1: Thema: Einführung in die formale Behandlung von Algorithmen (4 BE)

Beweisen Sie, dass Binärsuche eine Laufzeitverhalten von O(log n) hat, indem Sie

i) die Rekursionsgleichung aufstellen,

ii) die zu zeigende Ungleichung für die Rekursionsgleichung angeben,

iii) die Ungleichung aus ii) mit vollständiger Induktion beweisen.
Hinweis: In Ihrem Beweis dürfen Sie davon ausgehen, dass n eine 2-Potenz ist.

Aufgabe 2: Thema: Einführung in die formale Behandlung von Algorithmen (2 BE)

Sortieren Sie die folgenden Komplexitätsklassen:

a) $O(n^3 (\log_2 n)^2)$

b) $O(n^2 (\log_3 n)^3)$

c) $O(2^n)$

d) $O(n^{3,01})$

e) $O(n^{0,001})$

f) $O((\log_2 n)^{1000})$

g) $O((\log_2 (n^{1000}))$

Aufgabe 3: Thema: Einführung in die formale Behandlung von Algorithmen (4 BE)

Begründen Sie die folgenden Aussagen in Worten, wobei Sie nicht auf Details der Algorithmen eingehen müssen. Benutzen Sie c als Konstante für die obere Schranke und d für die untere Schranke.

a) Quicksort ist ein $\Theta(n^2)$-Algorithmus im schwachen Sinn, aber nicht im starken.

b) Selectionsort ist ein $\Theta(n^2)$-Algorithmus im starken Sinn.

Aufgabe 4: Thema: Such- und Sortieralgorithmen (6 BE)

Zeigen Sie den ersten Rekursionsschritt in der optimalen Lösung für das Auswahlproblem Select (5,A) für folgende Eingabe:

A = [19, 7, 1, 4, 18, 6, 24, 15, 9, 8, 3, 25, 11, 20, 5, 23, 17, 10, 14, 12, 16, 22, 13, 21, 2]

Geben Sie konkret alle Zwischenresultate bis zum nächsten rekursiven Aufruf des <u>ursprünglichen</u> Problems an (der rekursive Aufruf zur Bestimmung des Medians der Mediane soll berechnet werden, aber nur durch direkte Angabe der Lösung). Geben Sie die Parameter des nächsten rekursiven Aufrufs konkret an, also das nächste k und das nächste A.

Aufgabe 5: Thema: Such- und Sortieralgorithmen (4 BE)

Simulieren Sie Radixsort an folgendem Beispiel:

Tunnel, Bahn, Sonne, Boom, Turm, Bann, Sahne

Setzen Sie voraus, dass die Wörter in der hier vorgegebenen Reihenfolge eingegeben werden und geben Sie die Zwischenstände nach den einzelnen Sortierungsschritten an.

Aufgabe 6: Wörterbuchproblem (5 BE)

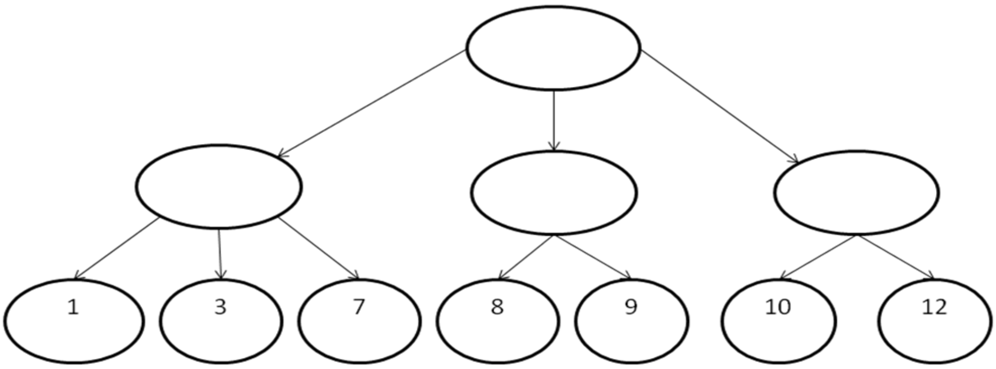
Gegeben sei der unten angegebene (2,3)-Baum.

a) Geben Sie die Schlüssel für die inneren Knoten im Beispiel an.

b) Löschen Sie das Element 10 mit einer rekursiven Prozedur delete23, welche die Enkel neu verteilt und dafür neue Kinder generiert. Zeichnen Sie den neuen Baum.

c) Fügen Sie dann das Element 2 neu hinzu mit einer Prozedur insert23, welche einen aufgerufenen Knoten aufspaltet, wenn er 4 Kinder bekommt und einen oder 2 Knoten an den Aufrufer zurückgibt. Zeichnen Sie den neuen Baum.

Hinweis: Die hier geschilderten Bedingungen entsprechen genau den Verfahrensweisen aus der Vorlesung.



Aufgabe 7: Thema: Graphenalgorithmen (4 BE)

Analysieren Sie die Laufzeit des Algorithmus von Dijkstra: Geben Sie im Code für jeden Schritt die Einzellaufzeit an und ermitteln Sie daraus die Gesamtlaufzeit.

Algorithm of Dijkstra for SSSP: (for graphs G with nonnegative edge costs only)

- Initialize the node set `Done` by s;
  Initialize the node set `Undone` by all other nodes of graph G;
  For all nodes v of the graph G:
  - Let label (v) := length of edge from s to v (∞ if no edge is existing, 0 if v = s);
- While `Undone` is not empty:
  - Search and delete the node v from `Undone` with minimal label;
  - Insert v into `Done`;
  - Update all neighbors n of v that are in `Undone`:
    - If label (n) > label (v) + length of edge between v and n:
      - Replace label (n) by that number;
      - Let v be the predecessor of n.

Aufgabe 8: Thema: Graphenalgorithmen (4 BE)

Betrachten Sie den Algorithmus von Floyd-Warshall (siehe unten):

a) Geben Sie an, welches Problem der Algorithmus löst.

b) Definieren Sie die Bedeutung von $d_{ij}^{(k)}$.

c) Geben Sie die Laufzeit an und begründen Sie diese anhand des Codes, indem Sie die Zwischenlaufzeiten daneben schreiben.

for $i = 1, \ldots, n$ **do**
    for $j = 1, \ldots, n$ **do**
        $d_{ij}^{(0)} = \begin{cases} c(i,j): & \text{falls } (i,j) \in E \\ \infty: & \text{sonst} \end{cases}$
    end for
end for
for $k = 1, \ldots, n$ **do**
    for $i = 1, \ldots, n$ **do**
        for $j = 1, \ldots, n$ **do**
            $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$
        end for
    end for
end for

Aufgabe 9: Thema: Graphenalgorithmen (2 BE)

Welche Schwierigkeit tritt bei der Berechnung eines maximalen Matchings in einem allgemeinen Graphen auf? Geben Sie an, mit welcher Technik der Algorithmus von Edmonds diese Schwierigkeit löst.

Aufgabe 10: Thema: String Matching (7 BE)

a) Berechnen Sie die Präfixfunktion π (i) für den Knuth-Morris-Pratt-Algorithmus für folgendes Eingabemuster:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	a	b	b	a	b	b	b	a	b	b	a	b	b	a	b	b	b	a	b	b	a	b	b	b
π (i)																								

b) Schildern Sie in Worten, wie der Algorithmus von Knuth-Morris-Pratt diese Präfixfunktion nutzt, um alle Matchings zu finden. Begründungen müssen nicht abgegeben werden. Definieren Sie dafür auch die genaue Problemstellung.

Aufgabe 11: Thema: Algorithmische Geometrie (6 BE)

Betrachten Sie das Problem, die konvexe Hülle von Punkten in der Ebene zu berechnen.

a) Wie löst man das Problem trivialerweise (keine genauen Formeln verlangt), und wie hoch ist die Laufzeit?

b) Wie kann man das Problem mit Hilfe eines Voronoidiagrams lösen, und wie hoch ist jetzt die Laufzeit (inklusive der Berechnung des Voronoidiagrams)?

Aufgabe 12: Thema: Probleme und Algorithmen (2 BE)

Betrachten Sie die Funktion auf den natürlichen Zahlen, die jeder Zahl den Wert ihrer maximalen Ziffer in Dezimaldarstellung zuordnet.

a) Definieren Sie zu diesem Berechnungsproblem das zugehörige Entscheidungsproblem, indem Sie die zugehörige Sprache in Worten oder formal bescheiben.

b) Geben Sie ein konkretes Element an, das zu der in a) definierten Sprache dazugehört und eines, das nicht dazugehört. Im zweiten Fall soll aber immer noch ein Wort über dem richtigen Alphabet und der richtigen Datenstruktur genommen werden.

Aufgabe 13: Thema: Berechenbarkeit und Nichtberechenbarkeit (6 BE)

Betrachten Sie folgende Funktion:

Der Input sei eine natürliche Zahl in unärer Darstellung.

Der Output sei die Zahl 1 in unärer Darstellung, wenn der Input gerade ist, und 0, wenn er ungerade ist.

Benutzen Sie die folgende Notation für die Turingmaschine: Die natürlichen Zahlen seien mit so vielen Strichen dargestellt, wie es ihrem Wert entspricht (kein Strich für 0). Das Defaultzeichen für alle Positionen des unendlichen Bandes sei *. Am Anfang stehe der LS-Kopf auf dem linkesten Strich der Eingabe (* falls die Eingabe 0 ist). Am Ende soll er auf der Ausgabe stehen. Die Bewegungsanweisungen seien L, S (für Stehenbleiben) und R.

a) Geben Sie jeweils ein Beispiel für Input und Output auf dem Turingband für eine gerade und eine ungerade Zahl an.

b) Geben Sie die Turingmaschine mit Zuständen und Übergangsfunktion an, welche diese Funktion berechnet.

Aufgabe 14: Thema: Berechenbarkeit und Nichtberechenbarkeit (2 BE)

Skizzieren Sie in Worten, wie man beweist, dass es Probleme geben muss, die nicht berechenbar sind. Nennen Sie vor allem das entscheidende Beweisprinzip, wonach man hier vorgeht.

Aufgabe 15: Thema: Berechenbarkeit und Nichtberechenbarkeit (6 BE)

Betrachten Sie die Sprache L über dem Alphabet aller 26 Buchstaben (groß und klein wird nicht unterschieden, keine Umlaute und Sonderzeichen), die aus allen Wörtern besteht, die wenigstens 2 verschiedene Zeichen enthalten.

a) Beschreiben Sie in Worten das Komplement von L und geben Sie jeweils ein Beispiel für ein Element aus L und aus seinem Komplement an.

b) Ist das Komplement von L berechenbar? Begründen Sie Ihre Antwort mit einem entsprechenden Satz für die Nichtberechenbarkeit oder der verbalen Beschreibung der Turingmaschine in Worten, welche diese Sprache erkennt.

c) Ist die Menge aller Gödelnummern, die zu Turingmaschinen gehören, die alle Wörter der Sprache L akzeptieren, berechenbar? Begründen Sie Ihre Antwort mit einem entsprechenden Satz für die Nichtberechenbarkeit oder der verbalen Beschreibung der Turingmaschine in Worten, welche diese Menge erkennt.

Aufgabe 16: Thema: Theorie der NP-Vollständigkeit (1 BE)

Was ist der einzige Unterschied in der formalen Definition zwischen DTM und NDTM?

Aufgabe 17: Thema: Theorie der NP-Vollständigkeit (2 BE)

Betrachten Sie die Situation, dass bereits bewiesen wurde, dass 3-SAT NP-vollständig ist, und Sie jetzt zeigen wollen, das VERTEX COVER (VC) auch NP-vollständig ist.

a) Stellen Sie mit Hilfe des Symbols $\propto$ dar, welche Reduktion zu zeigen ist.

b) Beschreiben Sie die Reduktionsaufgabe in Worten.

Aufgabe 18: Thema: Theorie der NP-Vollständigkeit (5 BE)

Das Problem CLIQUE hat als Eingabe einen Graphen G und eine natürliche Zahl k und beantwortet die Frage, ob es in G einen vollständigen Untergraphen mit k Ecken gibt.

a) Geben Sie an, wie man auf einer NDTM die Lösbarkeit von CLIQUE bestimmen kann und welche Laufzeit man dabei erzielt.

b) Begründen Sie, warum man das Problem CLIQUE für ein konstantes k auch auf einer DTM in polynomialer Laufzeit lösen kann. Geben Sie die asymptotische Laufzeit explizit an.

c) Warum können Sie mit dem Verfahren von b) CLIQUE nicht für beliebige k in polynomialer Laufzeit auf einer DTM lösen?