Höhere Algorithmik, WS 2014/15 — 15. Übungsblatt

20 Betonpunkte. Schriftliche Abgabe in Zweiergruppen bis Montag, 9. Februar 2015

81. (0 Punkte) Betrachten Sie die UNION-FIND-Datenstruktur mit Darstellung der Mengen als Bäume und Pfadkompression, und zwar mit (i) Vereinigung nach Rang, (ii) Vereinigung nach Größe, und (iii) Vereinigung in beliebiger Reihenfolge. Nehmen Sie an, dass auf einer Grundmenge von n Elementen, die zunächst Einzelmengen bilden, (a) alle k UNION-Operationen ($k \leq n-1$) *vor* allen m FIND-Operationen (b) alle m FIND-Operationen *vor* allen k UNION-Operationen stattfinden. In welchen dieser sechs Fälle ist die Gesamtlaufzeit linear, also $O(m+k)$, nachdem die Datenstruktur in $O(n)$ Zeit initialisiert wurde?

82. Fluten eines Tagbaus, digitale Landschaft, 10 Punkte

Ein rechteckiges $(m \times n)$-Feld $a[i,j]$ gibt die Höhe einer Landschaft in Abhängigkeit von den Koordinaten i und j an, siehe nebenstehendes Beispiel. Wenn das Grundwasser bis zur Höhe h steigt, werden die Bereiche mit $a[i,j] < h$ überflutet, und es entstehen Seen und Inseln. Ein See oder eine Insel muss dabei über gemeinsame Kanten zusammenhängen: Das heißt, das Quadrat $a[i,j]$ ist nur mit vier anderen Quadraten $a[i \pm 1, j]$ und $a[i, j \pm 1]$ benachbart, aber zum Beispiel nicht mit $a[i \pm 1, j \pm 1]$.

91	24	56	69	59	76	67
92	47	77	58	39	72	53
10	82	15	17	65	57	49
45	75	93	61	2	71	37
51	66	90	73	28	78	12
25	63	55	30	33	43	81
79	19	23	46	99	11	41
44	36	94	6	1	26	98
87	7	18	27	89	74	22

(a) (0 Punkte) Finden Sie alle bei Höhe $h = 50$ überfluteten Gebiete im obigen Beispiel, und malen Sie die Seen und Inseln in verschiedenen Farben an.

(b) (10 Punkte) Wir möchten die Entwicklung der Seen und Inseln bei wachsendem Wasserstand verfolgen, und insbesondere zu jeder Höhe h die Anzahl der Inseln und Seen wissen.

Entwerfen Sie einen effizienten Algorithmus für dieses Problem, der die Quadrate $a[i,j]$ in nach ihrem Wert sortierter Reihenfolge betrachtet und die lokalen Änderungen, die sich durch die Überflutung einer einzelnen Zelle ergeben, bearbeitet. Nehmen Sie eine UNION-FIND-Datenstruktur zu Hilfe. Nehmen Sie der Einfachheit halber an, dass alle Werte $a[i,j]$ verschieden sind. Analysieren Sie die Laufzeit.

Ähnliche Fragen treten auch bei der Bildverarbeitung auf; dort bedeutet $a[i,j]$ den Grauwert eines Bildpunktes (Pixels).

83. Produktionsplanung, 0 Punkte

Die Firma Zopel schätzt für die Monate des nächsten Jahres den Absatz an Knaster gemäß der nebenstehenden Tabelle ein. Als Produktionsleiter/in von Zopel können Sie in jedem Monat entscheiden, ob Sie Knaster herstellen oder nicht. Die Produktionskosten sind 50 Taler pro Tonne, plus zusätzlich 400 Taler Fixkosten, falls in diesem Monat überhaupt Knaster hergestellt wird. Der Knaster, der am Ende des Monats nicht gemäß der nebenstehenden Tabelle verkauft wird, kann zwischengelagert und später abgesetzt werden. Das Lagern kostet aber 17 Taler pro Tonne pro Monat. Beispiel: Wenn Sie alle 120 benötigten Tonnen Knaster im Januar produzieren, kostet das $400 + 120 \times 50 + 17 \times (108 + 105 + 104 + \cdots + 12)$ Taler.

Jan	12
Feb	3
Mär	1
Apr	15
Mai	8
Jun	25
Jul	2
Aug	11
Sep	20
Okt	9
Nov	2
Dez	12

Berechnen Sie die optimale Herstellungs- und Lagerungsstrategie. Das Lager ist zu Beginn leer. Der Bedarf muss auf jeden Fall erfüllt werden. Entwerfen Sie einen Algorithmus, der auch für einen Zeithorizont von mehreren Jahren geeignet ist.
Hinweis: Sie können auch von hinten nach vorne rechnen.

84. Felder variabler Länge (Vektoren), 0 Punkte

Ein Feld variabler wachsender Länge n kann implementiert werden, indem man es in ein Feld mit statischer Länge $n' \geq n$ speichert und ab und zu auf ein neues Feld mit geändertem n' umspeichert. Betrachten Sie folgenden Algorithmus, der das Wachsen von n unterstützt:

> Die Größe n' ist immer eine Zweierpotenz. Bevor n größer als n' wird, wird n' verdoppelt, und das Feld wird auf ein neues Feld der Größe n' kopiert.

Zeigen Sie mit Hilfe der Potentialfunktion $\Phi = 2n - n'$, dass das Vergrößern von n um 1 in konstanter amortisierter Zeit ausgeführt werden kann.

85. (0 Punkte) Schauen Sie nach, wie die Längenänderung in der Java-Standardklasse `java.util.ArrayList` implementiert ist.

86. Die Ackermannfunktion, 10 Punkte

Eine Variante der sogenannten Ackermannfunktion $A_i(n)$ ist für $i \geq 0, n \geq 1$ so definiert:

$$A_i(n) := \begin{cases} n+1, & \text{für } i = 0 \\ 3, & \text{für } i = 1,\ n = 1 \\ 2, & \text{für } i \geq 2,\ n = 1 \\ A_{i-1}(A_i(n-1)), & \text{für } i \geq 1,\ n > 1 \end{cases}$$

(a) (6 Punkte) Welche bekannten Funktionen sind A_1, A_2, A_3, A_4?

(b) (4 Punkte) Welche Werte stehen in den "Spalten" $A_i(1)$ und $A_i(2)$, $(i \geq 0)$?

(c) (0 Punkte) Betrachten Sie die beiden Umkehrfunktionen

$$\alpha(n) := \min\{\, i \geq 3 \mid A_i(3) \geq n \,\} = 2 + \min\{\, j \geq 1 \mid L_j(n) \leq 3 \,\}$$
$$\tilde{\alpha}(n) := \min\{\, i \geq 2 \mid A_i(i) \geq n \,\} = 2 + \min\{\, j \geq 0 \mid L_j(n) \leq j + 1 \,\}$$

(Die Funktionen $L_j(r)$ wurden in der Vorlesung definiert.[1]) Zeigen Sie, dass für alle n die Abschätzung $\tilde{\alpha}(n) \leq \alpha(n) \leq \tilde{\alpha}(n) + 1$ gilt.

87. (0 Punkte) Beweisen Sie, dass Spieler B das Ratespiel von Aufgabe 56 vom 9. Übungsblatt in $O(\alpha(n))$ Fragerunden gewinnen kann, wobei α in Aufgabe 86c definiert ist. (Diese Schranke ist asymptotisch optimal.)

88. Gradfolgen, 0 Punkte

Gibt es einen Graphen mit 5 Knoten mit Knotengraden $1, 1, 2, 3, 4$? Mit Knotengraden $1, 1, 2, 3, 3$? Mehrfachkanten und Schleifen sind dabei nicht erlaubt. Gibt es einen Graphen mit 6 Knoten mit Knotengraden $1, 2, 2, 3, 4, 4$? Mit Knotengraden $1, 1, 2, 2, 3, 3$? Mit Knotengraden $1, 1, 2, 3, 4, 5$?

Finden Sie einen Greedy-Algorithmus, der einen Graphen mit gegebener Gradfolge konstruiert, wenn es einen gibt. Beweisen Sie, dass es keinen Graphen gibt, wenn der Algorithmus versagt. (Schwierig)

Beweisen Sie, dass es für jedes n und jedes $d < n$ einen *d-regulären* Graphen (wo alle Knoten Grad d haben) auf n Knoten gibt, wenn dn gerade ist.

Was ändert sich, wenn Mehrfachkanten zugelassen werden? Werden die Probleme dadurch schwieriger oder leichter?

[1] Siehe `http://w3.inf.fu-berlin.de/lehre/WS14/HA/union-find.pdf`, Abschnitt 4.3