Algorithmen und Datenstrukturen

Aufgabe 1 (AGS 2.2.53 ⋆)

(a) Sei $\Sigma = \{a, b, c\}$. Geben Sie die Mengen V und R einer EBNF-Definition $\mathcal{E} = (V, \Sigma, S, R)$ an, so dass $W(\mathcal{E}) = \{\, a^k b^\ell c^{2k+m} \mid k \geq 1, \ell \geq m \geq 0 \,\}$ gilt.

(b) Sei $\Sigma = \{a, b\}$ und $\mathcal{E} = (V, \Sigma, X, R)$ eine EBNF-Definition mit $V = \{X, Y\}$ sowie

$$R = \{\ X ::= \hat{(} aXa \,\hat{|}\, Y \hat{)}\,, \quad Y ::= \hat{[} bY \hat{]}\ \}.$$

Geben Sie die für die Fixpunktsemantik von $\mathcal{E}$ zu iterierende Funktion f an. Dokumentieren Sie vier Iterationsschritte der Fixpunktsemantik von $\mathcal{E}$. Geben Sie die Sprache $W(\mathcal{E})$ an, die durch die EBNF-Definition $\mathcal{E}$ beschrieben wird.

Aufgabe 2 (AGS 3.2.13 ⋆)

Gegeben sei die folgende Typdefinition für einfach verkettete Listen in C:

```c
struct node { int wert; struct node *next; };
```

Von einer solchen Liste l soll ein vollständiger binärer Baum *full*(l) wie rechts dargestellt erzeugt werden:

```
2                 2
|               /   \
7    full      7     7
     |-->     / \   / \
|
4            4   4 4   4
```

- Die Höhe von *full*(l) entspricht der Länge von l.
- Jedem Knoten in *full*(l) mit der Entfernung $k \geq 0$ von der Wurzel, wird das k-te Listenelement als Beschriftung zugeordnet.

(a) Geben Sie eine geeignete Datenstruktur für binäre Bäume an.

(b) Geben Sie eine Funktion `full` in C an, die eine bestehende Liste l als Argument nimmt, den Baum *full*(l) konstruiert und einen Zeiger darauf zurückgibt.

(c) Geben Sie einen Funktionsaufruf von `full` an. Definieren Sie ggf. die benutzten Variablen.

Aufgabe 3 (AGS 6.1.13 ⋆)

Wenden Sie den Quicksort-Algorithmus auf die Folge $5, 9, 3, 8, 2$ an. Die Zahlen sollen aufsteigend sortiert werden.

Aufgabe 4 (AGS 6.2.15 ⋆)

Wenden Sie auf die Folge: $1, 2, 5, 8, 20, 14, 21, 17, 23, 25, 3, 16$ den Heapsort-Algorithmus an. Sie müssen in Phase zwei nur zwei Doppelschritte ausführen!

https://www.orchid.inf.tu-dresden.de thomas.ruprecht@tu-dresden.de

Aufgabe 5 (AGS 4.26 ⋆)

Gegeben sei das folgende C-Programm:

```
1  #include <stdio.h>
2
3  void f(int* x, int y) {
4    /* label1 */
5    *x = *x - 2;
6    if (*x > y)
7      f(x, y + 3); /*$1*/
8    /* label2 */
9  }
10 void g(int* a, int b) {
11   /* label3 */
12   while (b < *a) {
13     f(a, b - 1); /*$2*/
14     b = b + 1;
15     /* label4 */
16   }
17 }
18
19 int main() {
20   int m, n;
21   m = 6;
22   scanf("%i", &n);
23   /* label5 */
24   g(&n, m); /*$3*/
25   /* label6 */
26   return 0;
27 }
```

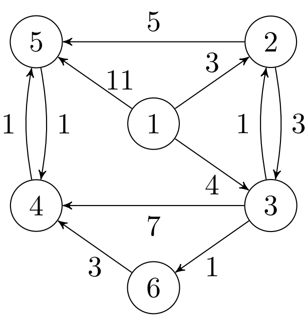
(a) Geben Sie den Gültigkeitsbereich jedes Objektes im Programm an. Nutzen Sie dazu die Zeilennummern.

(b) Setzen Sie das folgende Speicherbelegungsprotokoll fort. Dokumentieren Sie die aktuelle Situation beim Passieren der Marken (`label1` bis `label6`). Geben Sie jeweils den Rücksprungmarkenkeller und die *sichtbaren* Variablen mit ihrer Wertebelegung an. Die Inhalte von Speicherzellen nicht sichtbarer Variablen müssen Sie nur bei Änderungen eintragen. Die bereits festgelegten Rücksprungmarken sind `$1` bis `$3`.

Label	Rücksprungmarken	1	2	3	4	5	6	7	8	9	10
`label5`	–	`m` 6	`n` 14								

Aufgabe 6 (AGS 9.5.16 ⋆)

Der kantenbewertete Graph G sei durch folgende graphische Darstellung gegeben:



(a) Berechnen Sie mithilfe des Dijkstra-Algorithmus die minimale Entfernung vom Knoten mit der Nummer 1 zu allen erreichbaren Knoten. Protokollieren sie schrittweise die aktuelle Randknotenmenge und den zugehörigen gewählten Knoten. Geben Sie abschließend für alle berechneten kürzesten Wege die Entfernung und die jeweils zu durchlaufende Knotenfolge (Pfadtabelle) an.

(b) Welches ist das kleinste Gewicht, das man der Kante $(1, 2)$ zuweisen kann, so dass die unter 6 (a) berechneten *kürzesten Pfade* weiterhin gültig sind?

Aufgabe 7 (AGS 7.1.17 ⋆)

(a) Bestimmen Sie die mit Hilfe des KMP-Algorithmus berechnete Verschiebetabelle für das Pattern `aabaaabc`.

(b) Mit Hilfe des KMP-Algorithmus ist die unten stehende unvollständige Verschiebetabelle berechnet worden. Vervollständigen Sie das aus den Symbolen `a` und `b` bestehende Pattern.

https://www.orchid.inf.tu-dresden.de *thomas.ruprecht@tu-dresden.de*

Position	0	1	2	3	4	5
Pattern	`a`					
Tabelle	-1	0	-1	0	-1	3

Aufgabe 8 (AGS 7.2.5 $\star$)

Gegeben seien die Wörter $w =$ `espen` und $v =$ `beispiele`.

(a) Berechnen Sie die Levenshtein-Distanz $d(w, v)$. Geben Sie dazu die Berechnungsmatrix an. Tragen Sie alle Zelleneinträge zusammen mit den dazugehörigen Pfeilen ein.

(b) Geben Sie die Levenshtein-Distanz $d($`espe`, `beispiel`$)$ an. Beachten Sie, dass `espe` und `beispiel` Präfixe von `espen` bzw. `beispiele` sind.

(c) Geben Sie zwei Alignments zwischen `espen` und `beispiele` an, die zu den minimalen Kosten führen. Dabei sollen die Alignments die jeweils angewendeten Editieroperation enthalten.

(d) Wieviele Backtraces enthält die in Aufgabenteil 8 (a) angegebene Berechnungsmatrix?

Aufgabe 9 (AGS 10.19 $\star$)

Die Personen A und B spielen ein Spiel mit einer Münze mit den Beschriftungen 1 und 2, sowie mit einem dreiseitigen „Würfel“ mit den Beschriftungen 1, 2, und 3. In jeder Runde werden die Münze und der Würfel geworfen. Der Spieler A gewinnt die Runde, falls die gefallene Zahl des Würfels kleiner gleich der auf der Münze ist. Ansonsten gewinnt B die Runde. Die Menge der möglichen Ergebnisse ist somit $X = \{1, 2\} \times \{1, 2, 3\}$.

(a) Geben Sie den Analysator A für dieses Szenario an.

Die Spieler spielen 31 Runden des Spiels und A gewinnt dabei 21 Mal.

(b) Geben Sie den Korpus h mit unvollständigen Daten an.

Wir wollen aus diesem Korpus h mit dem EM-Algorithmus die Wahrscheinlichkeitsverteilungen von Münze und Würfel bestimmen. Die initiale Wahrscheinlichkeitsverteilung $q_0 = q_0^M \times q_0^W$ ist gegeben durch $q_0^M(1) = 1/3$ und $q_0^W(1) = q_0^W(3) = 1/4$. Dabei ist q_0^M die Wahrscheinlichkeitsverteilung der Münze, und q_0^W die des Würfels.

(c) Geben Sie q_0 an.

(d) Führen Sie den E-Schritt aus. Vervollständigen Sie also den Korpus h zum Korpus h_1.

(e) Führen Sie nun den M-Schritt aus. Bestimmen Sie dafür zunächst die Teilkorpora h_1^M und h_1^W für die Münze bzw. den Würfel.

(f) Schätzen Sie nun die Wahrscheinlichkeitsverteilung q_1^M der Münze sowie q_1^W des Würfels, indem Sie die relative Häufigkeit der Teilkorpora bestimmen.