Universität Stuttgart – Institut für Computerphysik

Neue Aufgaben

Computergrundlagen WS 2013/2014

JP Dr. Axel Arnold Dr. Olaf Lenz Florian Weik
Georg Rempfer Rudolf Weeber Tobias Richter
Shervin Rafatnia Johannes Zeman Dominic Röhm

27. Februar 2014

- Insgesamt können in der Klausur 100 Punkte erzielt werden (in der vergangenen Jahren nur 60!).
- Eine Aufgabe sollte zur Bearbeitung ungefähr eine Minute pro Punkt in Anspruch nehmen.
- Zum Bestehen der Klausur sind 50 Punkte notwendig.
- In der Klausur gibt es die folgenden Aufgabenblöcke und die dazugehörigen (ungefähren) Punktzahlen:
  1. Unixgrundlagen (20 Punkte)
  2. Python (15 Punkte)
  3. C (15 Punkte)
  4. Algorithmen und Datenstrukturen (30 Punkte)
  5. LaTeX und HTML (20 Punkte)
  6. Visualisierung (5 Punkte)

1 Algorithmen und Datenstrukturen

Hinweis Im folgenden ist mit *abstrakter Datenstruktur* die Benutzersicht auf eine Datenstruktur gemeint, also etwa eine Liste, eine assoziative Liste oder ein Wörterbuch. Mit *Datenstruktur* ist die Implementation einer abstrakten Datenstruktur gemeint, also etwa eine Hashmap, eine verkettete Liste oder ein Array.

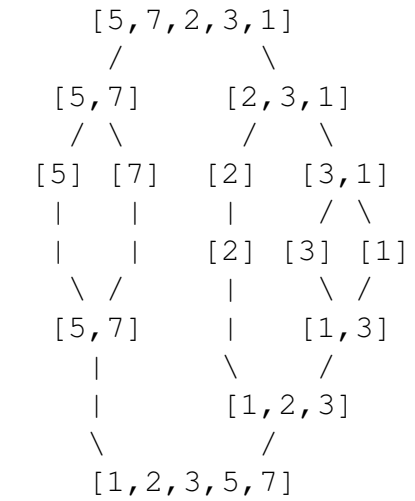
Aufgabe 1: (1 Punkt)
Was ist eine rekursive Funktion?
Antwort:

> Eine rekursive Funktion ist eine Funktion, die sich selbst aufruft.

Aufgabe 2: (2 Punkte)
Wie kannst Du ein absteigend sortiertes Array möglichst effizient aufsteigend sortieren?
Antwort:

> Symmetrisch von außen nach innen die Elemente vertauschen.

Aufgabe 3: (4 Punkte)
Verwende den Mergesort-Algorithmus, um die Liste `[5,7,2,3,1]` zu sortieren. Schreibe dabei die einzelnen Schritte des Zerlegens und des Wiederzusammenführens der Listen auf.
Antwort:



Aufgabe 4: (4 Punkte)

Beschreibe, wie der Quicksort-Algorithmus funktioniert!

Antwort:

1. Wenn die zu sortierende Liste nur ein Element enthält, dann gib die Liste zurück
2. Wähle eine Pivotelement aus der Liste (z.B. das erste Element)
3. Teile die Liste in zwei Unterlisten, bei der die eine Liste alle Elemente enthält, die kleiner als das Pivotelement sind, die andere Liste alle anderen Elemente.
4. Wende den Algorithmus rekursiv auf die beiden Unterlisten an.
5. Füge die (jetzt sortierten) Unterlisten und das Pivotelement in der richtigen Reihenfolge zusammen
6. Gib die Ergebnisliste zurück

Aufgabe 5: (3 Punkte)

Gegeben sei eine Liste von Wörtern, die in einer der folgenden Datenstukturen gespeichert ist: Sortierte einfach verkettete Liste, Sortiertes Array, binärer Suchbaum. Sortiere die Datenstrukturen absteigend nach der benötigten mittleren Zeit zum Zugriff auf das $n$te Element in aufsteigender Reihenfolge.

Antwort:

1. sortiertes Array (direkter Zugriff auf das $n$te Element, $\mathcal{O}(1)$)
2. sortierte einfach verkettete Liste (
3. binärer Suchbaum

Aufgabe 6: (4 Punkte)

Welche Datenstruktur (Array, einfach verkettete Liste, Hashmap, Binärbaum) würdest Du verwenden, wenn Du eine abstrakte Datenstruktur implementieren solltest, die sich wie ein Warteschlange verhält. Eine Warteschlange zeichnet sich dadurch aus, dass Elemente effizient auf der einen Seite zugefügt und effizient auf der anderen Seite wieder entfernt werden können. Warum würdest Du genau diese Struktur verwenden, und keine andere?

Antwort:

Verkettete Liste mit einem Zeiger auf Kopf und Schwanz, um effizient Löschen und Einfügen zu können.

Aufgabe 7: (4 Punkte)

Was tut die folgende C-Funktion vermutlich?

```
Element *function(int value, Element *pos) {
  Element *element = (Element *)malloc(sizeof(Element));
  element->value = value;
  element->next = pos->next;
  pos->next = element;
  return element;
}
```

Antwort:

Einfügen von `value` in eine einfach verkettete Liste.

Aufgabe 8: (7 Punkte)

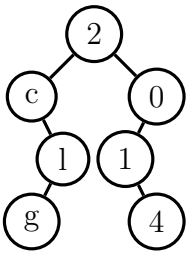
Ein Palindrom ist ein Wort, das von hinten gelesen dasselbe Wort ergibt wie von vorne, beispielsweise das Wort *Lagerregal*. Schreibe eine C-Funktion, die für eine beliebige Zeichenkette zurückgibt, ob diese ein Palindrom ist oder nicht.

Antwort:

```
int ispalindrome(char *word) {
  int a = 0;
  int b = strlen(word)-1;
  while (a < b) {
    if (word[a] != word[b]) return 0;
    a++; b--;
  }
  return 1;
}
```

Aufgabe 9: (2 Punkte)

Was ist die Ausgabe eines Programms, das den folgenden Binärbaum in symmetrischer Traversierung (linker Teilbaum, Knoten, rechter Teilbaum) ausgibt?



Antwort:

cgl2140

Aufgabe 10: (5 Punkte)

Schreibe eine C-Funktion, die den Inhalt eines binären Suchbaums aufsteigend sortiert ausgibt. Ein Knoten des Binärbaumes sei definiert wie folgt:

```
struct node {
  int value;
  struct node *left, *right;
};
```

Die kleineren Elemente seien dabei stets im linken Ast.

Antwort:

```
void print_tree(struct node* tree) {
  if (tree == NULL) return;
  print_tree(tree->left);
  printf("%d\n", tree->value);
  print_tree(tree->right);
}
```

2 LaTeX und HTML

Aufgabe 11: (5 Punkte)

Übersetze das folgende LaTeX-Dokument in ein entsprechendes HTML-Dokument

```
\documentclass[a4paper]{scrartcl}
\begin{document}
\section{Verhaltensregeln}
In einer Klausur darf man folgende Dinge nicht tun:
\begin{itemize}
\item \textbf{Abschreiben} (und sich dabei erwischen lassen)
\item \emph{Laut} singen
\end{itemize}
\end{document}
```

Antwort:

```
<html>
<head></head>
<body>
<h1>Verhaltensregeln</h1>
In einer Klausur darf man folgende Dinge nicht tun:
<ul>
<li><strong>Abschreiben</strong>
(und sich dabei erwischen lassen)
<li><em>Laut</em> singen
</ul>
</body>
</html>
```

Aufgabe 12: (2 Punkte)

Für welche der Elemente des LaTeX-Dokuments gibt es *keine* direkte Entsprechung in HTML?

Antwort:

Der `\documentclass`-Befehl hat keine Entsprechung in HTML.