

Übungen zu Algorithmen und Datenstrukturen Aufgabenblatt 9

Prof. Dr. Steffen Staab,

Dipl.-Inform. Gerd Gröner und Dipl.-Inform. Christoph Ringelstein
Arbeitsgruppe ISWeb, Fachbereich Informatik, Universität Koblenz-Landau

Abgabe: 21.01.2009, bis 23:59 Uhr

1 Hashing (8 Punkte)

- Implementieren Sie die Java-Klasse `VerketteteListe`, die eine verkettete unsortierte Liste implementiert.
- Implementieren Sie die Java-Klasse `HashArray`. Diese Klasse soll ein Array von verketteten Listen verwalten. Die Länge des Arrays ist n und soll als Parameter des Konstruktors der Klasse übergeben werden.
- Implementieren Sie die Member-Methode `void einfuegen (int i)` der Klasse `HashArray`. Diese Methode soll das Element i in die passende verkettete Liste unter Verwendung von `InsertionSort` einfügen. Die passende Liste wird dabei durch die Hashfunktion $h(i) = i \bmod n$ bestimmt. Treten Duplikate auf sollen diese ebenfalls eingefügt werden.
- Implementieren Sie das `InsertionSort` als Teil der Methode `einfuegen`.
- Implementieren Sie die Member-Methode `boolean istEnthalten (int i)` der Klasse `HashArray`. Diese Methode soll überprüfen, ob ein Element mit dem Wert i in der Hashtabelle enthalten ist. Ist i enthalten, soll `true` (ansonsten `false`) zurückgegeben werden.
- Implementieren Sie die Member-Methode `int berechneLoadFactor ()` der Klasse `HashArray`, die den Load-Factor berechnet.

Die angegebenen Signaturen dürfen nicht verändert werden. Achten Sie darauf, dass Ihr Programm sehr gut kommentiert ist!

2 Erweitertes Hashing (4 Punkte)

Zeigen Sie an den folgenden Werten beispielhaft das erweiterte Hashing. Nutzen Sie den umgekehrten Bitvektor des Wertes zur Adressierung. Fangen Sie bei einer Tiefe von $d = 1$ des Tries an und stellen Sie diesen als Array dar. Die Blockgröße ist 2. Geben Sie für jeden Einfügeschritt jeweils den Index sowie die zugehörigen Blöcke mit ihrem Inhalt an (analog zum Beispiel aus der Vorlesung). Die Werte sind:

- 58
- 21
- 57
- 15
- 36
- 19
- 29

Fügen Sie diese bitte in der gegebenen Reihenfolge ein.

Wichtige Hinweise:

- Die SVN-Repositories sind unter <http://lehre2.isweb.uni-koblenz.de/svn/aud-ws0809/> zu erreichen, gefolgt vom jeweiligen Gruppennamen.
- Bitte checken Sie Ihr Programm und Ihre Texte und Grafiken ins SVN-Repository ein. Nutzen Sie als Verzeichnisname direkt unterhalb Ihres Gruppenverzeichnisses *zwingend*:

`blatt09`

Nur so können die Korrektoren Ihre Implementierung aus dem SVN-Repository auslesen und die Punkte verteilen.

- Achten Sie bitte darauf, dass Ihr Programm fehlerfrei übersetzt werden kann. Die Korrektoren werden Ihren Quellcode ebenfalls übersetzen und ausführen.
- Es werden nur Programme berücksichtigt, die in Java implementiert sind und direkt kompiliert werden können.
- Programme müssen gut kommentiert sein.
- **Bitte checken Sie KEINE .class-Dateien ein!**
- Wenn Sie Grafiken oder Textdokumente mit abgeben möchten, so verwenden Sie bitte als Dateiformat:
 - Für schematische Zeichnungen: .png oder .gif
 - Für Scans oder Photos: .png oder .jpg
 - Für Text: Plaintext (.txt) oder PDF-Dateien (.pdf). Natürlich dürfen Sie die Grafiken auch in eine PDF-Datei einbetten.

Andere Formate werden bei der Korrektur *nicht* berücksichtigt!

- Verwenden Sie bitte keine Umlaute, Leerzeichen oder Sonderzeichen in Dateinamen (auch nicht bei .pdf oder .png Dateien)!
- Verwenden Sie UTF-8 als Kodierung für Plaintext (.txt) und Java (.java) Dateien.