
Aufgabe 3.4 (WOERTERBUCHSTANDARD)

Lernziele

- ⇒ verkettete Strukturen
- ⇒ sortierter Binärbaum
- ⇒ rekursiver Algorithmus

Wiederholungsfragen

1. Welche Aussage ist *richtig*?

- a) In Datenfeldern müssen alle Elemente dieselbe Grösse haben und in einem zusammenhängenden Speicherbereich gespeichert werden.
- b) In verketteten Strukturen müssen alle Elemente dieselbe Grösse haben, sie müssen aber nicht in einem zusammenhängenden Speicherbereich gespeichert werden.
- c) In verketteten Strukturen können die Elemente verschiedene Grösse haben, sie müssen aber in einem zusammenhängenden Speicherbereich gespeichert werden.
- d) Sowohl in Datenfeldern als auch in verketteten Strukturen können die Elemente verschiedene Grösse haben, und sie müssen nicht in einem zusammenhängenden Speicherbereich gespeichert werden.

2. Welche Höhe hat ein ausbalancierter Binärbaum mit 31 Knoten?

- a) 30 b) 31 c) 32 d) 4

3. Bild 35 zeigt fünf Binärbäume. Welche sind sortiert?

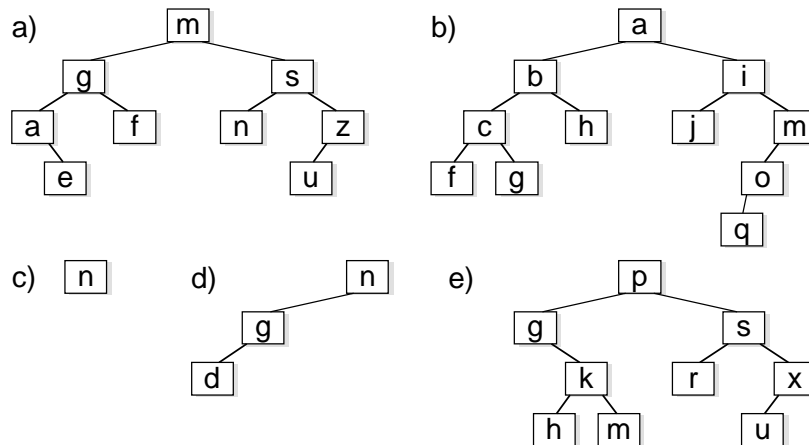
- a) b) c) d) e)

4. Bild 35 zeigt fünf Binärbäume. Welche sind ausbalanciert?

- a) b) c) d) e)

5. Skizzieren Sie die Binärbäume, die entstehen, wenn Sie folgende Buchstaben, in der angegebenen Reihenfolge in einen leeren sortierten Binärbaum einfüllen.

- a) x, v, s, r, o, e, n
- b) r, v, e, x, s, o, n
- c) n, o, s, x, e, v, r

**Bild 35:** Beispiele sortierter und unsortierter Binärbäume

6. Welche Aussage ist *falsch*? Bei einem rekursiven Algorithmus ...


- a) ... müssen die Abbruchbedingungen vor einem rekursiven Aufruf geprüft werden.
- b) ... muss der Basisfall die Rekursion abbrechen.
- c) ... muss die Reduktion vor dem Basisfall geprüft werden.
- d) ... muss der rekursive Aufruf die Aufgabe vereinfachen.

Vertiefungsfragen

Bevor wir mit der eigentlichen Aufgabe beginnen, möchten wir den Debugger noch etwas besser kennenlernen. Laden Sie hierfür die Arbeitsmappe [LverketteteListe.xls](#) aus dem Lösungsverzeichnis, und wechseln Sie in die Programmierungsumgebung.

a) Machen Sie sich mit dem Programmcode vertraut.

- Aus welchen Modulen setzt sich das Projekt zusammen?
- Welche Funktion hat das Modul *cZelle*?
- Welche Prozeduren implementiert das Modul *Prozeduren*, und wozu wird die Variable *Anfang* benötigt?

b) Wechseln Sie ins Modul *Prozeduren*, setzen Sie den Cursor auf die Variable *Anfang*, und klicken Sie auf die Schaltfläche  in der Debugger-Symbolleiste (oder drücken Sie <Umschalt> + <F9>). Bestätigen Sie den Dialog, der erscheint, mit der Schaltfläche 'Hinzufügen'. Sie können nun im Fenster Überwachungsausdrücke den Inhalt der Variablen *Anfang* bei der Programmausführung jederzeit ansehen.

- Platzieren Sie den Cursor auf die Prozedur Test und durchlaufen Sie die Prozedur mit Prozedurschritten (☞☞☞ oder <Umschalt> + <F8>). Beobachten Sie den Inhalt der Variablen Anzahl. Achten Sie darauf, dass Sie die Datenstruktur immer mit ☞ expandieren.
- Klicken Sie in der Debugger-Symbolleiste auf die Schaltfläche Zurücksetzen (☐), um die Variableninhalte wieder zu löschen.
- Fügen Sie die Variable suche in der Prozedur suche zu den Überwachungsausdrücken hinzu, und lassen Sie die Testprozedur noch einmal durchlaufen. Verwenden Sie diesmal Einzelschritte (☞☞☞ resp. <F8>), sobald die Prozedur suchen aufgerufen werden soll. Verfolgen Sie den Ablauf des Suchalgorithmus.

Wir möchten nun eine einfache Variante unseres Wörterbuchs 'Deutsch-Englisch' für Fachbegriffe implementieren. Bild 36 zeigt die Benutzerschnittstelle aus der Arbeitsmappe ☞[WoerterbuchStandardSkelett.xls](#).

- füge hinzu: Fügt einen neuen Eintrag (Begriff/Definitions-Paar) ins Wörterbuch ein.
- übersetze: Gibt die Definition zu einem Begriff zurück.

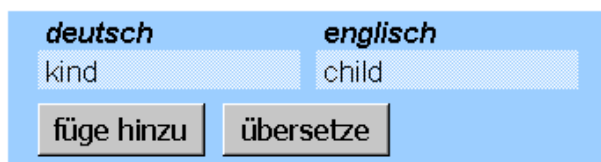


Bild 36: Benutzerschnittstelle der Aufgabe ☞WoerterbuchStandard

Aufgaben:

- Öffnen Sie die Arbeitsmappe ☞[WoerterbuchStandardSkelett.xls](#), und machen Sie sich mit ihrem Inhalt vertraut:
 - Welche Namen haben die hellblauen Bereiche?
 - Welche Makros sind den beiden Schaltflächen zugeordnet?
 - Aus welchen Modulen setzt sich das Projekt zusammen?
 - In welchem Modul befinden sich die Ereignisprozeduren?
 - Welche Aufgaben haben die restlichen Module?

b) Wechseln Sie in die Programmierungsumgebung, und betrachten Sie das Klassenmodul cWörterbuch:

- Notieren Sie alle Eigenschaften und Methoden der Klasse cWörterbuch.
- Beschreiben Sie den Unterschied zwischen den drei Prozeduren fügeHinzu(), fügeHinzuIt() und fügeHinzuRek(). Welche Unterschiede in den Prozedurköpfen stellen Sie fest?
- Beschreiben Sie den Unterschied zwischen den beiden Prozeduren Eintrag() und EintragRek(). Welche Unterschiede in den Prozedurköpfen stellen Sie fest?
- Schreiben Sie den Programmcode der Prozedur erzeugeKindRe().

c) Wechseln Sie ins Modul Prozeduren, und vervollständigen Sie den Programmcode:

- Beschreiben Sie die Funktion der Prozedur initialisiere(), und fügen Sie die fehlende Programmzeile ein.
- Beschreiben Sie die Funktion des Makros fügeHinzu_Klick(), und ergänzen Sie den Programmcode.
- Beschreiben Sie die Funktion des Makros übersetze_Klick(), und ergänzen Sie den Programmcode.

d) Testen und studieren Sie die beiden Implementationen des Einfügealgorithmus.

- Wechseln Sie ins Modul cWörterbuch, und überzeugen Sie sich, dass die Methode fügeHinzu() die iterative Version des Einfügealgorithmus aufruft. Setzen Sie beim Aufruf einen Haltepunkt, und legen Sie einen Überwachungsausdruck für die Variable Wurzel fest.
- Geben Sie über die Benutzerschnittstelle die folgenden Begriffs/-Definitions-Paare ein. Beobachten Sie dabei den Programmablauf mit dem Debugger und den Aufbau der Datenstruktur im Überwachungsfenster.

kind/child, ast/branch, wurzel/root, knoten/node.

Welche Bedingung bricht den Algorithmus in den folgenden Fällen ab?

- i) Wenn Sie das erste Mal ein Begriffspaar eingeben.
 - ii) Wenn Sie ein weiteres Begriffspaar eingeben.
 - iii) Wenn Sie ein Begriffspaar zwei Mal eingeben.
- Lassen Sie nun die Methode fügeHinzu() die rekursive Version des Einfügealgorithmus aufrufen, und beobachten Sie den Programmablauf erneut.

- e) Entwerfen und implementieren Sie eine rekursive Prozedur `EintragRek()`, die einen Begriff im Wörterbuch sucht und den gefundenen Eintrag zurück gibt.
- Schauen Sie sich zunächst den Aufruf der Prozedur `EintragRek()` in der Methode `Eintrag()` an. Weshalb wird die Suche nicht direkt in der Methode `Eintrag()` implementiert?
 - Notieren Sie ein paar Beispielbäume, die sich zum Testen des Algorithmus eignen.
 - Entwerfen Sie einen rekursiven Algorithmus für die Suche eines Begriffs im Baum (vgl. Bild 34).
 - Implementieren Sie den Algorithmus.
 - Testen Sie ihren Algorithmus anhand ihrer Beispielbäume. Geben Sie dazu über die Benutzerschnittstelle einen (deutschen) Begriff ein und klicken anschliessend auf 'übersetzen'.

✓ [LWoerterbuch.xls](#)

FügeHinzuRekursiv Begriff, Baum

'1. Basisfall

FALLS Baum leer

 erzeuge Wurzelknoten

 fülle Begriff ein.

'2. Reduktion

SONST

 FALLS Begriff < Begriff im Wurzelknoten von Baum

FügeHinzuRekursiv Begriff, linker Teilbaum von Baum.

 FALLS Begriff > Begriff im Wurzelknoten

FügeHinzuRekursiv Begriff, rechter Teilbaum von Baum.

Bild 34: rekursiver Einfügealgorithmus
