Informatik I WS 2005/2006

Übungsblatt 12

Abgabe: 8.2.06 - 10.2.06

Aufgabe 1 (ADT, Dictionaries) Punkte: 5

Aus der Vorlesung kennen Sie bereits den Python-Datentyp `dictionary` als Menge von Key-Value-Paaren. Formalisiert lässt er sich folgendermaßen ausdrücken:

Die Menge der Sorten:

- D = Menge der Dictionaries.
- V = Menge der Werte
- K = Menge der Keys
- B = Menge der Wahrheitswerte

Die zugehörige Signatur:

- new: $\rightarrow$ D
- insert: D $\times$ K $\times$ V $\rightarrow$ D
- delete: D $\times$ K $\rightarrow$ D
- read: D $\times$ K $\rightarrow$ V
- isempty: D $\rightarrow$ B

Geben Sie einen vollständigen Satz von Axiomen an und Klassifizieren Sie die Operationen.

Aufgabe 2 (Suchen) Punkte: 5

Vergleichen Sie die lineare, binäre und exponentielle Suche sowie Interpolation Search miteinander. Implementieren Sie dazu die Suchalgorithmen in Python und fügen Sie Ihre Implementation in das Framework zur Laufzeitmessung ein, das auf der Vorlesungs-Homepage bereit steht. Messen Sie die Zeit, die die Algorithmen zur Suche in unterschiedlich großen, sortierten Floating-Point-Arrays benötigen.

Erstellen Sie eine Tabelle mit den Laufzeiten für verschiedene Arraygrößen und ermitteln Sie auf Ihrem Rechner:

- Wann ist die binäre Suche schneller als die lineare Suche?
- Wann ist die Interpolation Search schneller als die binäre Suche?
- Wann ist die exponentielle Suche schneller als die binäre Suche?

Aufgabe 3 (Komplexität) Punkte: 3+3

a) Ordnen Sie folgende Funktionen in eine Folge f_1, f_2, f_3, f_4, so daß gilt: $f_i \in O(f_{i+1})$.

- $(3/2)^n$
- n^3
- $(\log n)^{(\log n)}$
- $4^{\log n}$

b) Welche der folgenden Aussagen sind wahr/ sinnvoll?

1. (a) $O(n) \in O(n^2 \log n)$
   (b) $n^2 \log n \in O(n^2 \log n)$
   (c) $O(\log n) \in O(n^2 \log n)$
   (d) $n^2 \in O(n^2 \log n)$
2. (a) $n \log n \in O(n)$
   (b) $n \log n \in \Omega(n)$
   (c) $n^2 \in O(n)$
   (d) $n^2 \in \Omega(n)$
3. (a) $\Omega(n^4) \supseteq O(n^4)$
   (b) $O(n^4) \supseteq O(n^4)$
   (c) $\Omega(n^3) \supseteq O(n^4)$
   (d) $O(n^3) \supseteq O(n^4)$

Begründen Sie Ihre Antworten ausführlich.

Tip: Bei manchen Aussage genügt die Angabe eines Gegenbeispiels.

Aufgabe 4 (Komplexität)

Punkte: 4

Berechnen Sie die Laufzeit $T(n)$ des folgenden Algorithmus:

```
mmult(m1,m2):
       m2dim = len(m2)
       m1dim = len(m1)
       if m1dim != m2dim: raise IndexError, "matrices don't match"
       result = [ None ] * m1dim
       for i in range( m1dim ):
           result[i] = [0] * m2dim
           for j in range( m1dim ):
              for k in range( m1dim ):
                 result[i][j] = result[i][j] + m1[i][k] * m2[k][j]
       return result
```

Der Algorithmus berechnet das Produkt zweier quadratischer Matrizen m1 und m2. Die Matrizen liegen als 2-dimensionale-Listen vor.

Dieser Aufgabenzettel wird in der letzten Übungsstunde, also in der Übungsstunde, in der er auch abgegeben wird, sofort besprochen. Deswegen muss die Abgabe zu Beginn der Übungsstunde erfolgen. Auch wenn Sie die Punkte von diesem Übungszettel nicht mehr benötigen um das 30%/50%-Scheinkriterium zu erfüllen, sollten Sie die Aufgaben trotzdem unbedingt bearbeiten, denn wie immer können Aufgaben dieser Art in der Vordiplomsklausur vorkommen.