

Praktikum 1: Sortieren, Polynom-Evaluation und Inversionen

Präsentation: 15.10. u. 17.10.2003

Name: Matr-Nr:

Datum: Unterschrift des Dozenten (wenn bestanden):

Hinweis: Literatur zu den in den Aufgaben behandelten Themen finden Sie in allen Büchern über Datenstrukturen und Algorithmen.

Aufgabe 1 (InsertSort kombiniert mit MergeSort)

- a) Zeige, dass n/k Teillisten jeweils der Länge k durch InsertSort im Worst Case in $\Theta(nk)$ Zeit sortiert werden kann.
- b) Zeige, dass die Teillisten in $\Theta(n \log(n/k))$ Worst-Case-Zeit gemischt werden können.
- c) Gegeben ist ein modifizierter Algorithmus, der in $\Theta(nk + n \log(n/k))$ Worst-Case-Zeit läuft. Welches ist der größte asymptotische (Θ -Notation) Wert für k als Funktion von n , für den der modifizierte Algorithmus dieselbe asymptotische Laufzeit hat wie der Standard-MergeSort-Algorithmus?
- d) Wie würde man k in der Praxis wählen?

InsertSort(A)

```
for j ← 2 to length[A]
  do key ← A[j]
    // Füge A[j] in die sortierte Folge A[1..j-1] ein
    i ← j-1
    while i > 0 and A[i] > key
      do A[i+1] ← A[i]
        i ← i-1
    A[i+1] ← key
```

Aufgabe 2 (Evaluation von Polynomen)

Ein Polynom $P(x) = \sum_{k=0}^n a_k x^k$

$$= a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + xa_n)\dots))$$

kann für einen Wert x und Koeffizienten a_0, a_1, \dots, a_n nach der Horner-Regel evaluiert werden. Der Algorithmus dazu lautet im Pseudocode:

```

1   y ← 0
2   u ← n
3   while i ≥ 0
4       do y ← ai + x * y
5       i ← i-1

```

- Welche asymptotische Laufzeit hat der Algorithmus für die Horner-Regel?
- Schreiben Sie einen Algorithmus in Pseudocode, der das Polynom auf primitive Art und Weise evaluiert, d.h. jeder Term des Polynoms wird komplett berechnet. Welche Laufzeit hat dieser Algorithmus? Wie arbeitet er im Vergleich zur Horner-Regel?
- Prüfe folgende Schleifeninvariante der While-Schleife:
- Am Anfang jeder Iteration gilt: $y = \sum_{k=0}^{n-(i+1)} a_{k+i+1} x^k$
- Erklären Sie, dass das gegebene Codefragment das Polynom für gegebene Koeffizienten a_0, a_1, \dots, a_n korrekt evaluiert

Aufgabe 3 (Inversionen)

Sei $A[1..n]$ ein Array mit n verschiedenen Zahlen. Wenn $i < j$ und $A[i] > A[j]$, dann heißt das Paar (i, j) eine **Inversion** von A .

- Wie lauten die 5 Inversionen der Folge $\langle 2, 3, 8, 6, 1 \rangle$?
- Welches Array mit Elementen aus der Menge $\{1, 2, \dots, n\}$ hat die meisten Inversionen? Wie viele Inversionen sind es?
- Welche Beziehung besteht zwischen der Laufzeit von InsertSort und der Anzahl der Inversionen des Eingabefeldes? Begründung!
- Geben Sie einen Algorithmus an, welcher die Anzahl der Inversionen in einer beliebigen Permutation von n Elementen in $\Theta(n \log n)$ worst-case-Zeit bestimmt. Hinweis: modifiziere MergeSort.