Algorithmen & Datenstrukturen

Blatt 3

Dr. Matthias Thimm
Tina Walber, Leon Kastler, Martin Leinberger und Maximilian Strauch
Fachbereich Informatik, Universität Koblenz-Landau

7. Dezember 2013

1 Sortieren per Hand (4 Punkte)

Sortieren Sie die angegebenen Folgen jeweils mit den Sortieralgorithmen MergeSort und QuickSort, die in der Vorlesung vorgestellt wurden. Notieren Sie dabei die Parameter mit denen jeweils die Funktionen `mergeSort` und `quickSort` aufgerufen werden. Beachten Sie hierbei auch die rekursiven Aufrufe. In der Funktionen `quickSort` wird das mittlere Element als Pivot Element gewählt.

1. [2,9,4,5]
2. [4,7,9,1,2,8,3]

a) `mergeSort([2,9,4,5])` (1 Punkt)

b) `mergeSort([4,7,9,1,2,8,3])` (1 Punkt)

c) `quickSort([2,9,4,5])` (1 Punkt)

d) `quickSort([4,7,9,1,2,8,3])` (1 Punkt)

2 Stabilität bei QuickSort (5 Punkte)

Wie wir aus der Vorlesung wissen, ist QuickSort als Sortieralgorithmus nicht stabil.

1. Erklären Sie den Begriff der Stabilität.
2. Beweisen Sie, dass QuickSort nicht stabil ist (Hinweis: Beweis durch Gegenbeispiel).
3. Entwicklen Sie einen Ansatz um QuickSort stabil zu machen.

3 CocktailSort (5 Punkte)

Im folgenden Beispiel untersuchen wir den Sortieralgorithmus CocktailSort[1] (auch ShakerSort genannt)[2]. Der Algorithmus ist dem BubbleSort sehr ähnlich, allerdings bewegt er sich nicht von jeweils nur einer Seite der Liste zur anderen, sondern wechselt die Richtung am jeweiligen Ende.

1. Welchen Aufwand hat der Algorithmus im schlechtesten Fall, wenn man die Anzahl der Vertauschungen betrachtet?
2. Geben Sie ein Beispiel-Array an, der beim CocktailSort diesen schlechtesten Fall darstellt.
3. Zeigen Sie, wie in der Vorlesung, wie es zu diesem Aufwand für den schlechtesten Fall kommt.

Hinweis:

`swap(A,i,j)` tauscht das i-te Element vom Array A mit dem j-ten.

```
void cocktailSort(int[] A) {:
   do {
      swapped = false

      for(int i = 0; i < A.size() - 1; i++) {
         if (A[i] > A[i+1]) {
            swap(A, i, i+1);
            swapped = true;
         }
      }

       if (!swapped){
        break;
       }

      swapped = false;

       for(int i = A.size() - 1; i >= 0 ; i--) {
         if (A[i] > A[i+1]) {
             swap(A, i, i+1);
              swapped = true;
         }
      }
   } while (swapped);
}
```

[1] http://en.wikipedia.org/wiki/Cocktail_sort
[2] http://de.wikipedia.org/wiki/Shakersort

4 Wahl des Pivotelements bei QuickSort (6 Punkte)

In der Aufgabe 1 wurde das Pivotelement auf das mittlere Element der Liste gelegt: `int p = (u + o) / 2;`
Es kann aber auch das erste oder letzte Element einer Folge als Pivotelement gewählt werden. Die Wahl des Pivotelements kann die Effizienz des QuickSort-Algorithmus beeinflussen (siehe Foliensatz 07). Implementieren Sie den QuickSort Algorithmus und untersuchen Sie die Auswirkungen der Wahl des Pivotelements. Es kann aber auch das erste oder letzte Element einer Folge als Pivotelement gewählt werden. Die Wahl des Pivotelements kann die Effizienz des QuickSort-Algorithmus beeinflussen. Implementieren Sie den QuickSort Algorithmus und untersuchen Sie die Auswirkungen der Wahl des Pivotelements.

1. Implementieren Sie dazu zunächst eine abstrakte Klasse `Sort`, welche die abstrakte Methode `execute(int[] a)` definiert. Entwickeln Sie dann eine Klasse QuickSort, welche die Klasse Sort implementiert (`QuickSort extends Sort`) und den QuickSort Algorithmus beinhaltet. (1 Punkt)

2. Implementieren Sie drei Varianten von QuickSort, in denen die unterschiedlichen Wahlmöglichkeiten des Pivotelements (erstes Element, mittleres Element oder letztes Element der Folge) umgesetzt werden. (2 Punkte)

3. Programmieren Sie einen Zähler, der zählt wie oft der Algorithmus rekursiv aufgerufen wird. (1 Punkt)

4. Berechnen Sie die Anzahl der Rekursionsschritte für jede der drei Versionen von Quicksort für die Folgen (1 Punkt):
   a) a_1 = [3,1,2,0,9,6,5,8,4,7]
   b) a_2 = [0,1,2,3,4,5,6,7,8,9]
   c) a_3 = [9,8,7,6,5,4,3,2,1,0]

5. Interpretieren Sie die Ergebnisse! Welche Version des Algorithmus ist besonders effektiv bzw. ineffektiv für welche Zahlenfolge? (1 Punkt)