

## 6. Übung – Informatik I für VIW

### Fakultät Verkehrswissenschaften

### Fachrichtung Verkehrsingenieurwesen

Zeitraum: 10.01. bis 21.01.2011 (WS 2010/11)

#### Aufgabe 1 – Insertion Sort

(a) Sortieren Sie die folgenden drei Zahlenfolgen mit Hilfe des Insertion-Sort-Verfahrens. Zeigen Sie bei jedem Sortierschritt, wo die Grenze zwischen sortierten und unsortierten Elementen liegt, welches Element einsortiert wird, und welche sortierten Elemente um eine Position nach rechts verschoben werden müssen.

- 5, 2, 9, 1, 6
- 3, 1, 6, 8, 2, 7
- 8, 5, 4, 7, 10, 3, 6

(b) Der größte Aufwand beim Insertion-Sort-Algorithmus liegt im Verschieben von bereits sortierten Elementen; beim Sortieren der Folge 3, 2, 5, 1 sind zum Beispiel vier solche Verschiebungen nötig (eine Verschiebung beim Einsortieren von 2 und drei Verschiebungen beim Einsortieren von 1). Geben Sie eine Folge der Länge sechs an, bei deren Sortierung genau acht solche Verschiebung nötig sind. Gibt es eine Folge der Länge sechs, bei der sechzehn Verschiebungen notwendig sind?

#### Aufgabe 2 – Mergesort

(a) Sortieren Sie die folgenden zwei Zahlenfolgen mit Hilfe des Mergesort-Verfahrens. Stellen Sie den Sortierprozess ähnlich wie in der Vorlesung dar.

- 8, 1, 3, 5, 9, 6, 4, 2
- 2, 8, 10, 4, 15, 5, 3, 11, 1, 6, 16, 7, 9, 12, 14, 13

(b) Wir nennen zwei sortierte Folgen *verzahnt*, wenn beim Mergen der beiden Folgen immer abwechselnd ein Element der zweiten und ein Element der ersten Folge entfernt wird; die beiden Folgen 3, 7, 10, 12 und 2, 4, 9, 11 sind zum Beispiel verzahnt. Bestimmen Sie eine Folge der Länge 8 (bestehend aus den Zahlen 1 bis 8), so dass beim Sortieren dieser Folge bei jedem Mergevorgang nur verzahnte Folgen „aufeinandertreffen“.

#### Aufgabe 3 – Quicksort

(a) Sortieren Sie die folgenden zwei Zahlenfolgen mit Hilfe des Quicksort-Verfahrens. Stellen Sie jeden auftretenden Zerlegungsprozess detailliert dar, wobei Sie jeweils die zu zerlegende Teilfolge und die Indizes  $i$  und  $j$  in mindestens den folgenden Situationen notieren:

i) vor der Zerlegung, ii) vor jeder Vertauschung und iii) nach erfolgter Zerlegung. Beachten Sie, dass (abhängig vom Wert der Indizes  $i$  und  $j$  nach der Zerlegung) das Pivot-Element entweder zur ersten Teilfolge, zur zweiten Teilfolge oder zu keiner der Teilfolgen zugefügt wird.

- 8, 9, 6, 2, 3, 7
- 9, 8, 4, 2, 3, 7, 10

(b) Bestimmen Sie eine Folge der Länge sechs, welche bei Anwendung des Quicksort-Algorithmus den Worst-Case zur Folge hat; das heißt, in jedem Zerlegungsschritt wird von der zu zerlegenden Teilfolge nur ein Element abgespaltet.

Geben Sie weiterhin eine Folge der Länge sechs an, bei der der Best-Case erreicht hat.

#### **Aufgabe 4 (Zusatz) – Bucketsort**

Das folgende Sortierverfahren kann nur bei der Sortierung von Zahlenfolgen verwendet werden, insbesondere für solche, in denen Elemente mehrfach vorkommen. Es heißt Bucketsort und läuft wie folgt ab:

- Stelle für jede Zahl  $n$ , die in der Folge vorkommt, einen Zähler bereit, diesen Zähler benennen wir  $\#n$ .
- Für jeden Zahl  $n$ , zähle wie oft  $n$  in der Eingabefolge vorkommt und merke dir diese Zahl in  $\#n$ .
- Gehe die Zähler der Reihenfolge nach durch. Für den Zähler  $\#n$  gib die Zahl  $n$   $m$ -mal aus, wobei  $m$  der Wert ist, der in  $\#n$  gespeichert ist.

Schreiben Sie eine C-Funktion für Bucketsort und bestimmen Sie die Laufzeitkomplexität unter der Annahme dass alle Zahlen in der Eingabefolge beschränkt sind (zum Beispiel kleiner als hundert).