Team: 9, Tim Hagemann, Tim Hartig

Aufgabenaufteilung:

1. quicksortRekursiv, quicksortRandom
2. Leistungserfassung (Messung)

Quellenangaben: http://me.dt.in.th/page/Quicksort/

Begründung für Codeübernahme: Es wurde kein Code Übernommen.

Bearbeitungszeitraum: 26.11.2014 09:00 – 10:30
03.12.2014 08:30 – 11:20
06.12.2014 18:00 – 18:45

Aktueller Stand: Skizze ist fertiggestellt und Verständnis ist vorhanden. quicksortRekursiv ist fertig implementiert.

QuickSort

Übersicht

Der QuickSort-Algorithmus arbeitet nach dem Prinzip „Teile-und-herrsche“ („divide and conquer“). Dabei wird ein sog. Pivot-Element ermittelt, das dazu dient, das Array zu partitionieren, d.h. es wird in drei Teile aufgeteilt (Alle Elemente kleiner als das Pivot, das Pivot selbst und alle Elemente größer-gleich als das Pivot). Das Pivot muss dazu an die dafür korrekte Position innerhalb des Arrays platziert werden, um zu gewährleisten, dass nach der Teilung alle Elemente, die kleiner als das Pivot sind, sich im linken Teil befinden, die Größeren oder Gleichen im rechten Teil. Für die Vorgabe der Aufgabenstellung wiederholen wir den Vorgang der Partitionierung für alle, sich durch die Partitionierung ergebenen, Teilarrays mit mehr als elf Elementen. Für Teilarrays mit gleich oder weniger als elf Elementen, verwenden wir einen Sortieralgorithmus aus der zweiten Praktikumsaufgabe (Selection Sort bzw. Insertion Sort).

Partitionierung

Zuerst wird ein Pivot-Element bestimmt (im Folgenden abgekürzt mit „Pivot“). Bei quicksortRekursiv wird das linkteste Element des (Teil-)Arrays als Pivot gewählt, bei quicksortRandom wird ein zufälliges Pivot aus dem (Teil-)Array gewählt. Dieses Pivot muss an der korrekten Stelle im (Teil-)Array positioniert werden. Dazu werden alle Elemente, außer dem Pivot, als zugedeckte Karten betrachtet. Nun werden alle verdeckten Karten von links nach rechts sukzessive aufgedeckt und mit dem Pivot verglichen. Ist die Karte größer als das Pivot oder gleich, bleibt die Karte aufgedeckt und es wird die nächste Karte aufgedeckt. Ist die Karte kleiner als das Pivot, wird sie mit der Ersten, der aufgedeckten Karten (falls vorhanden), getauscht und die kleinere Karte zugedeckt.
Falls sich das Pivot unter den offenen Karten befindet, wird es mit der ersten offenen Karte getauscht.
Falls sich das Pivot unter den geschlossenen Karten befindet, wird es mit der letzten geschlossenen

Karte getauscht.
Andernfalls befindet sich das Pivot bereits an der richtigen Stelle.

Nach der Partitionierung werden die sich ergebenen Teilarrays, wie in der Übersicht beschrieben, weiterverarbeitet.

Funktionssignaturen

quicksortRekursiv: array x pos x pos → array x messdaten

(Zu sortierendes Array x Startposition x Endposition) → (Sortiertes Array x Messdaten)

quicksortRandom: array x pos x pos → array x messdaten

(Zu sortierendes Array x Startposition x Endposition) → (Sortiertes Array x Messdaten)