
Übungsmaterial zu Teile und Herrsche

Florian Gnepper, Stephan Strehler, Hochschule Zittau/Görlitz

13.11.2014

1 Implementation Quicksort

Implementieren Sie Quicksort anhand des Pseudocodes in eine Programmiersprache Ihrer Wahl. Dabei muss Ihre Umsetzung nicht den Eigenschaften eines internen Suchverfahren entsprechen. Machen Sie Testmessungen. Nehmen Sie zum Testen auch Elementarfälle wie eine sortierte Liste.

```
function quicksort (liste):  
  quicksort(liste, 0, liste.laenge-1) ;  
  
function quicksort (liste, links, rechts):  
if rechts > links then  
  | pivot= partition(liste, links, rechts);  
  | quicksort(liste,links,pivot-1);  
  | quicksort(liste, rechts, pivot+1);  
end
```

```

function partition (liste, links, rechts):
pivot=liste[links];  indexVor = links+1;  indexZur = rechts;
while indexZur>indexVor do
    while indexVor ≤ indexZur und liste[indexVor] ≤ pivot do
        | indexVor=indexVor+1;
    end
    while indexVor ≤ indexZur und liste[indexZur] >pivot do
        | indexZur=indexZur-1;
    end
    if indexZur >indexVor then
        | parkplatz = liste[indexZur];
        | liste[indexZur]= liste[indexVor];
        | liste[indexVor] = parkplatz;
    end
end
while indexZur >links und liste[indexZur] ≥ pivot do
    | indexZur = indexZur+1;
end
if pivot >liste[indexZur] then
    | liste[links] = liste[indexZur];
    | liste[indexZur] = pivot;
    | return indexZur;
else
    | return links;
end

```

2 Implementation Mergesort

Implementieren Sie Mergesort anhand des Pseudocodes in der Programmiersprache, mit der Sie auch Aufgabe 1 umgesetzt haben. Machen Sie ebenfalls Testmessungen und vergleichen Sie die Ergebnisse. Vergleichen Sie das Resultat auch mit dem Vergleich aus der Vorlesung.

```

function mergesort ( liste ):
if liste.laenge ≤ 1 then
    | return liste;
else
    | linkeListe = liste[0] bis liste[laenge/2];
    | rechteListe = liste[laenge-(laenge/2)] bis liste[laenge-1];
    | linkeListe =mergesort(linkeListe);
    | rechteListe=mergesort(rechteListe);
    | return merge(linkeListe, rechteListe);
end

```

```

function merge ( linkeListe, rechteListe ):
indexLinks=0 ;
indexRechts=0 ;
indexErgebnis=0 ;
neueListe[linkeListe.laenge+rechteListe.laenge] ;
while indexLinks <linkeListe.laenge und indexRechts <rechteListe.laenge do
|   if linkeListe[indexLinks] <rechteListe[indexRechts] then
|   |   neueListe[indexErgebnis]=linkeListe[indexLinks];
|   |   indexLinks=indexLinks+1 ;
|   else
|   |   neueListe[indexErgebnis]=rechteListe[indexRechts];
|   |   indexRechts=indexRechts+1 ;
|   end
|   indexErgebnis=indexErgebnis+1;
end
while indexLinks = linkeListe.laenge do
|   neueListe[indexErgebnis] = rechteListe[indexRechts];
|   indexLinks=indexLinks+1;
|   indexErgebnis=indexErgebnis+1;
end
while indexRechts = rechteListe.laenge do
|   neueListe[indexErgebnis] = rechteListe[indexRechts];
|   indexRechts=indexRechts+1;
|   indexErgebnis=indexErgebnis+1;
end
return neueListe ;

```

3 Zusatz: Implementation von schneller Matrixmultiplikation

Entwerfen Sie ein Programm, mit welchem man 2 Matrizen multiplizieren kann. Das Programm soll Folgendes beherrschen:

- die Größen der Matrizen soll 4 betragen
- die Matrizen können per Zufall entstehen oder manuell eingegeben werden
- neben der berechneten Matrix soll auch die Zeit ausgegeben werden, die das Programm benötigt hat

Für diese Aufgabe gibt es kein Pseudocode oder sonstige Vorgaben. Nutzen Sie dafür Ihre Kenntnisse aus der Veranstaltung Software Engineering I.

Bemerkungen

Arbeiten Sie bitte das Aufgabenmaterial in der hier vorgegeben Reihenfolge ab. Uns ist durchaus bewusst, dass es schon Vorlagen für die Implementierung von Quicksort und Mergesort in Java im Internet gibt. Allerdings ist es nicht Sinn der Übung diese einfach zu kopieren, da so das Grundverständnis der Verfahren, besonders der rekursiven Aufrufe, nicht gebildet wird.

Die Aufgabe 3 ist als Zusatzaufgabe zu verstehen. Diese wird genutzt, um den Unterschied der Bearbeitungszeit auszugleichen. Das bedeutet, dass die Aufgabe nur von den Studenten bearbeitet werden soll, welche schon nach der Hälfte der Zeit mit den 2 Grundaufgaben fertig sind.