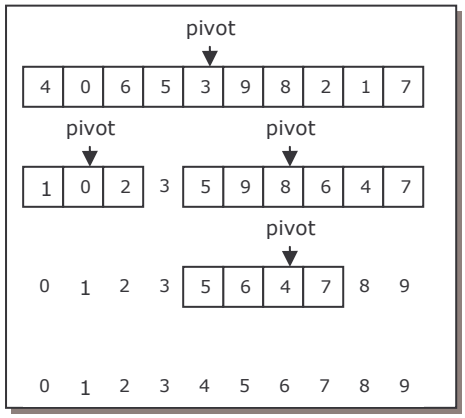
Übersichtsblatt:
Quick Sort

Idee:

Der Quick Sort ist ein Sortieralgorithmus, der nach dem Divide-And-Conquer Prinzip funktioniert. Die Idee ist, dass man ein übergebenes Feld in zwei Teilfelder zerlegt. Wobei man hier so vorgeht, dass man einen beliebigen Schlüssel (pivot) festlegt. Hierfür wird zumeist das mittlere Element benutzt aber auch das erst oder das letzte sind möglich. Das eine Teilfeld wird so gebildet, dass es alle Elemente enthält die kleiner sind als pivot, das andere enthält alle Elemente die größer sind. So wird das Feld Rekursiv immer weiter aufgeteilt bis nur noch einzelne Elmente übrig sind, dann setzt man aus diesen einfach wieder ein Feld zusammen.



Pseudocode:

Eingabe: Ein unsortiertes Feld F, untere Grenze links und obere Grenze rechts.
Ergebnis: Ein sortiertes Feld F', das alle Elemente aus F enthält.

```
int quickSost(F, links, rechts){
   l = links;
   r = rechts;
   if( r > l ){
      pivot = F[ (l + r) / 2 ];          /*Mitte als pivot verwenden!*/
      while( l <= r ){
         while( F[l] < pivot ){      /*Gehört Element in linke Folge?*/
            l++;
         }
         while( F[r] > pivot ){     /*Gehört Element in rechte Folge?*/
            r--;
         }
         if( l <= r ){            /*Wurden Tauschkandidaten gefunden?*/
            swap(F[l], F[r]);
            l++;
            r--;
        }
      }
   }
   if( links < r ){         /*Test ob linke Teilliste nicht leer ist!*/
      quickSort(F,links,r);
   }
   if( r < rechts ){      /*Test ob rechte Teilliste nicht leer ist!*/
      quickSort(F,l,rechts);
   }
   return F;
}
```

Analyse:

Eine Analyse des obigen Algorithmus ergibt, dass seine Laufzeit im Bereich

$O(n\log(n))$

liegt. Er kann allerdings auch ausarten, wenn z.B. eine Bereits sortierte Folge übergeben wird und als pivot nicht das mittlere sondern das erste Element verwendet wird.

Dann benöigt man n Rekursionsaufrufe. Das führt zu einer Laufzeit von

$O(n^2)$.

© Marco Rancinger

eTeachers.de
Lernen leicht gemacht