

Informatik OTG	Sortierverfahren	
	Quicksort	

Quicksort ist ein Sortierverfahren. Ein ausgewähltes Element ist an die ausgewählte Arrayposition zu bringen. Dazu sind Vergleiche notwendig.

11 03 21 42 08 07

11 sei das ausgewählte Element a_i . Es wird auch alselement bezeichnet. Bei der Suche von links ist die das erste Element > 11 . Diese wird in die linke Hand genommen. Nun beginnt von rechts die Suche bis zu dem Element, das kleiner (oder gleich) als 11 ist. Dieses - die - wird in die rechte Hand genommen. Nun werden und miteinander getauscht.

11 03 07 42 08 21

Nun wird wieder von links aus gesucht und zwar bei der Position, die im vorigen (linken) Durchlauf noch nicht betrachtet wurde. Das Element 42 wird in die linke Hand genommen und es wird nun von rechts gesucht, beginnend bei der Position, die noch nicht besucht wurde. Also wird 08 in die rechte Hand genommen und beide Elemente werden miteinander getauscht.

11 03 07 08 42 21

Wenn sich in der Fortsetzung dieses Prozesses linke mit rechter Suche kreuzt, wird das Element in der linken Hand zurückgelegt, das der rechten mit dem zu Beginn gewählten Element hier die 11 - getauscht.

08 03 07 11 42 21

Element 11 ist nun auf dem endgültigen Platz. Nacheinander wird nun das Verfahren für den Bereich links von der 11 und für den Bereich rechts von der 11 separat angewendet. (-> Rekursion)

07 03 08 11 21 42

03 07 08 11 21 42

Worst case: Zur Aufteilung eines Feldes der Länge N wird das Pivotelement mit den Schlüsseln aller Elemente im aufzuteilenden Bereich je einmal verglichen.

Ungünstigster Fall: alle Elemente je einmal ihren Platz

Pivotelement ist Element mit kleinstem / größtem Schlüssel: eine der beiden durch Aufteilung entstehenden Folgen ist jeweils leer, die andere hat jeweils ein Element weniger als die Ausgangsfolge

$$C_{\max}(n) \geq n + (n-1) + (n-2) + \dots + 1 = O(n^2).$$

max. Anzahl von Schlüsselvergleichen

ist von quadratischer Ordnung

$$M_{\max}(n) = O(n^2)$$

max. Anzahl von Bewegungen

Best case:

Teilfolgen sind stets etwa gleich lang.

Da damit der Baum der initiierten rekursiven Aufrufe die minimale Höhe $h = \log n$ hat, folgt:

$$C_{\min}(n) = O(n \log n)$$

Implementation in Delphi

```

procedure quicksort (links, rechts: Integer);
var i, j: Integer; x, w: Element;
begin
  if rechts > links then //mehr als ein Element enthalten?
  begin
    x := A[(links + rechts) div 2]; //Vergleichselement festlegen
    i := links;
    j := rechts;
    repeat
      while A[i] < x do i := i + 1; //Bewegung des linken Zeigers
      while A[j] > x do j := j - 1; //Bewegung des rechten Zeigers
      if i <= j then
        begin
          w := A[i]; //Vertauschen von A[i] und A[j]
          A[i] := A[j];
          A[j] := w;
          i := i + 1; //Zeigerbewegung
          j := j - 1;
        end;
      until i > j;
      quicksort (links, j);
      quicksort (i, rechts);
    end;
  end;

```