

Quicksort - Algorithmus (Pseudocode)

```

funktion quicksort(daten, links, rechts)
  falls links < rechts dann
    teiler := teile(daten, links, rechts)
    quicksort(daten, links, teiler-1)
    quicksort(daten, teiler+1, rechts)
  ende falls
ende function

funktion teile(daten, links, rechts)
  i := links
  j := rechts - 1      // Starte mit j links vom PivotElement
  pivot := daten[rechts]
  wiederhole solange wahr begin
    wiederhole solange daten[i] < pivot
      i := i + 1
    ende wiederhole
    wiederhole solange (links <= j) und (daten[j] > pivot)
      j := j - 1
    ende wiederhole
    falls i < j dann
      tausche daten[i] mit daten[j]
    sonst begin
      tausche daten[i] mit daten[rechts]
      return i
    ende
  ende wiederhole
ende function

```

Dijkstra-Verfahren (Pseudocode)

Knotenmenge K disjunkt in erkundete Knoten E , Grenzknoten G und unerkundete Knoten U unterteilen, d.h. $K = E \cup G \cup U$, $E \cap G = \emptyset$, $E \cap U = \emptyset$, $G \cap U = \emptyset$. $s \in K$ ist **Startnoten**.

Wegefolge $w(k)$ ist geordnete Knotenfolge von durch gewichtete Kanten verbundene Knoten (Nachbarknoten) von s nach $k \in K$ minimaler Länge.

Distanz $d(k) :=$ Länge des minimalen Weges $w(k)$, d.h. Summe der Kantengewichte von $w(k)$.

Anfang: $w(s) = s$, $d(s) = 0$, $G = \{s\}$, $U = K \setminus \{s\}$, $E = \emptyset$, $d(k) = \infty$ für $k \in U$

Algorithmenschritt, solange $G \neq \emptyset$:

Suche Knoten $k \in G$ mit $d(k) = \text{Minimum}(d(i), i \in G)$, E um k erweitern, G um Nachbarknoten $j \in U$ von k erweitern, Knoten j aus U und Knoten k aus G entfernen.

Für aus U in G aufgenommene Nachbarknoten j : $w(j)$ und $d(j)$ auf Basis von $w(k)$ und $d(k)$ bestimmen. Für Nachbarknoten $i \in E$ von k : $d^*(i) = d(k) + \text{gewicht}(k, i)$ mit $d(i)$ vergleichen und falls $d^*(i) < d(i)$, dann $d(i) := d^*(i)$ und $w(i) := w(k) \parallel (k, i)$, d.h. $w(k)$ mit Kante (k, i) verketten.