<u>func</u> <u>QuickSort(S)</u>

// Eingabe: Eine Menge S mit einer vollständigen Ordnung $\leq$

// Ausgabe: Die Elemente von S in sortierter Reihenfolge

// bezüglich $\leq$ (d. h. als Sequenz)

if $|S| \leq 1$ **then**

// Da gibt's nicht mehr viel zu sortieren...

return S;

fi

// Nicht-trivialer Fall: $|S| > 1$

Wähle (nach einer bestimmten Strategie) ein

Pivotelement $y \in S$;

Vergleiche y mit allen Elementen in $S' := S \setminus \{y\}$

und erzeuge zwei Mengen:

$S_1 := \{x \in S' \ : \ x < y\}$;

$S_2 := \{x \in S' \ : \ x > y\}$;

// Wende Quicksort rekursiv an.

L_1 := QuickSort(S_1);

L_2 := QuickSort(S_2);

// Die sortierte Reihenfolge von S ergibt sich durch

// Kombination der Teilergebnisse.

return $L_1 \circ y \circ L_2$;

Sortiere mit qsort() aus C-Library $n = 50.000$ Elemente:

```
Zufällig erzeugt:                1,08 Sekunden
Aufsteigend vorsortiert:         0,46 Sekunden
Absteigend vorsortiert:          0,44 Sekunden
Auf-/absteigend vorsortiert:    11,22 Minuten (!!)
```