function mergesort (A, p, r) **begin**

  if $p < r$ **then**

   $q \leftarrow \left\lfloor \frac{p+r}{2} \right\rfloor$ } (b)   Rekursion!

   mergesort (A, p, q) } (a)

   mergesort $(A, q+1, r)$ }

   merge (A, p, q, r) } (b)

  fi

end

Wie bestimmt man die Laufzeit von solchen Algorithmen?

~> Master theorem?

(1) Aufteilen in

  (a) Rekursive Aufrufe für Teilprobleme

  (b) Nicht-rekursiven Teil

(2) Laufzeit formulieren als Funktion $T(n): \mathbb{N} \to \mathbb{R}$, die die Eingabegröße auf eine Laufzeit abbildet

hier: $T(n) = \underbrace{\Theta(1)}_{\text{Ber. v. } q} + \underbrace{T(\tfrac{1}{2}n) + T(\tfrac{1}{2}n)}_{\text{2 rek. Aufrufe mit je halber Eingabegröße}} + \underbrace{\Theta(n)}_{\text{merge}}$

(3) Lösen mit Mastertheorem:

Sei $T(n): \mathbb{N} \to \mathbb{R}$ mit

$$T(n) = \underbrace{\sum_{i=1}^{m} T(\alpha_i n)}_{\text{m rekursive Aufrufe für Teilprobleme der Größe } \alpha_1 n, \alpha_2 n, \alpha_3 n, \dots, \alpha_m n} + \underbrace{\Theta(n^k)}_{\text{Zusammenfügen von Teillösungen}}$$

wobei $m \in \mathbb{N}$, $k \in \mathbb{R}$, $\forall i = 1, \dots, m$: $\underbrace{0 < \alpha_i < 1}_{\text{Teilprobleme müssen kleiner werden}}$

$$T(n) \in \begin{cases} \Theta(n^k) & \text{falls } \sum_{i=1}^{m} \alpha_i^k < 1 \\ \Theta(n^k \log n) & \text{falls } \sum_{i=1}^{m} \alpha_i^k = 1 \\ \Theta(n^c) & \text{falls } \sum_{i=1}^{m} \alpha_i^k > 1 \text{ für } \sum_{i=1}^{m} \alpha_i^c = 1 \end{cases}$$

Fall 1: ⇝ Merge auf Rekursionsebene „billiger" als auf Hauptebene ⇝ Hauptebene dominiert Laufzeit

Fall 2: ⇝ Haupt- und Rekursionsebene halten sich die Waage

Fall 3: ⇝ Rekursionsebene überwiegt

Zurück zum Bsp.: $T(n) = 2T(\frac{1}{2}n) + \Theta(n)$

(1) Parameter ermitteln: $k = 1$ (weil $\Theta(n) = \Theta(n^1)$)
$m = 2$ (2 rek. Aufrufe)
$\alpha_1 = \alpha_2 = \frac{1}{2}$ (halbierte Eingabegröße)

(2) Fallunterscheidung: $\sum_{i=1}^{m} \alpha_i^k = \left(\frac{1}{2}\right)^1 + \left(\frac{1}{2}\right)^1 = 1$ ⇝ Fall 2!

(3) Ergebnis ablesen: $T(n) \in \Theta(n^k \log n) \overset{k=1}{=} \Theta(n \log n)$

Bsp: $T(n) = 256\, T\left(\frac{n}{4}\right) + n^3$

$$= \sum_{i=1}^{256} T\left(\tfrac{1}{4} n\right) + \Theta(n^3)$$

$\alpha_1 = \dots = \alpha_{256} = \frac{1}{4}$

$m = 256$

$k = 3$

$$\sum_{i=1}^{m} a_i^k = \sum_{i=1}^{256} \left(\frac{1}{4}\right)^3 = 256 \cdot \frac{1}{64} \underline{>} 1$$

$\hookrightarrow$ Fall 3

$\rightsquigarrow$ suche c mit $\sum_{i=1}^{m} a_i^c = 1$

$$\Leftrightarrow \sum_{i=1}^{256} \left(\frac{1}{4}\right)^c = 1 \Leftrightarrow 256 \left(\frac{1}{4}\right)^c = 1$$

$$\Leftrightarrow \left(\frac{1}{4}\right)^c = \frac{1}{256} \Leftrightarrow 4^c = 256$$

$$\Leftrightarrow c = \log_4 256 \Leftrightarrow c = 4$$

$$\Rightarrow T(n) \in \Theta(n^c) = \Theta(n^4)$$

Bsp.: $T(n) = 27\,T\left(\frac{n}{3}\right) + n^3$

$\alpha_1 = \dots = \alpha_{27} = \frac{1}{3}$

$m = 27$

$k = 3$

$\sum_{i=1}^{m} \alpha_i^k = \sum_{i=1}^{27} \left(\frac{1}{3}\right)^3 = 27 \cdot \frac{1}{27} = 1$ $\rightarrow$ Fall 2

$\Rightarrow T(n) \in \Theta\left(n^k \log n\right) = \Theta\left(n^3 \log n\right)$

Bsp.: $T(n) = 3\,T\left(\frac{n}{4}\right) + n^2$

$\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{4}$

$m = 3$

$k = 2$

$\sum_{i=1}^{m} \alpha_i^k = \sum_{i=1}^{3} \left(\frac{1}{4}\right)^2 = 3 \cdot \frac{1}{16} = \frac{3}{16} < 1$ $\rightarrow$ Fall 1

$\Rightarrow T(n) \in \Theta\left(n^k\right) = \Theta\left(n^2\right)$

Quicksort

```
function QUICKSORT(A, p, r)
    if p < r then
        q ← PARTITION(A, p, r)
        QUICKSORT(A, p, q − 1)
        QUICKSORT(A, q + 1, r)
    end if
end function
```

<u>Idee:</u>

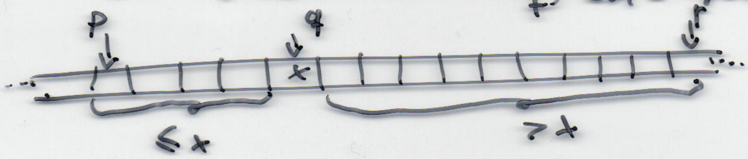
Feld ≤ 1 Elemente -> fertig

sonst: wähle Pivotelement x

alles ≤ x : links von x

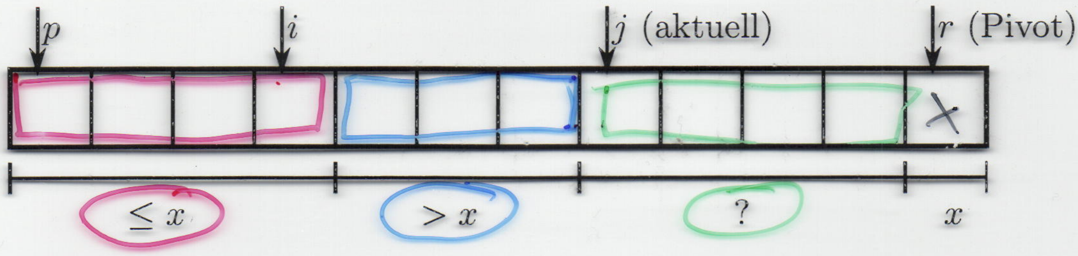
alles > x : rechts von x

(neuen) Index von x (q) zurückgeben




```
function PARTITION(A, p, r)
    x ← A[r]
    i ← p − 1
    for j ← p, . . . , r − 1 do
        if A[j] ≤ x then
            i ← i + 1
            SWAP(A[i], A[j])
        end if
    end for
    SWAP(A[i + 1], A[r])
    return i + 1
end function
```

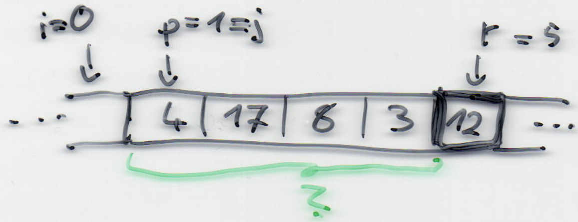
„In Place"



Bsp.: Wende partition auf $[4, 17, 8, 3, 12]$ an!

$A[1]$ ... $A[5]$

9-10:

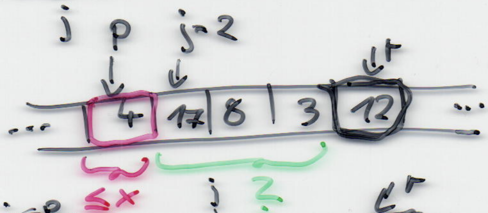


$x \leftarrow A[r] = 12$

$i \leftarrow p-1 = 0$

11-16:

$j=1$: $\underbrace{A[1]}_{j} \leq \underbrace{12}_{x}$ => Tausche $A[\underset{i+1}{1}] \leftrightarrow A[1]$

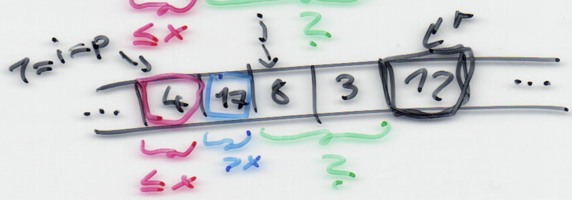


$i := i+1 = 1$

$j := j+1 = 2$

$j=2$: $A[\underset{j}{2}] = 17 > 12$ => Kein Tausch

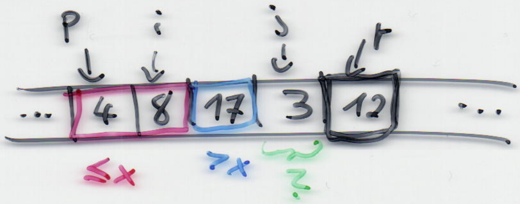
$j := j+1 = 3$



$j=3$: $A[\underset{j}{3}] = 8 \leq 12$ => Tausche $A[\underbrace{i+1}_{2}] \leftrightarrow A[\underbrace{j}_{3}]$

$i := i+1 = 2$

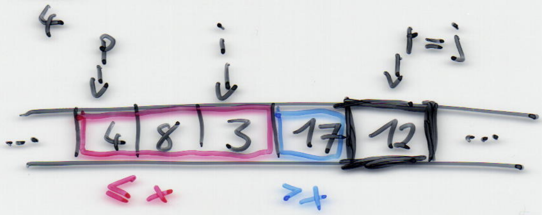
$j := j+1 = 4$



$j=4$: $A[\underset{j}{4}] = 3 \leq 12$ => Tausche $A[\underbrace{i+1}_{3}] \leftrightarrow A[\underbrace{j}_{4}]$

$i := i+1 = 3$

$j := j+1 = 5$



17-18: Zum Schluss: Pivotelement zwischen „≤x"-Bereich und „>x"-Bereich tauschen: Tausche $A[r] \leftrightarrow A[i+1]$

Rückgabewert: $q = i+1 = \underline{4}$