

DGA Ü2

1) k sortierte Listen $A[n(i-1)+1, \dots, n_i]$ für $i=1, \dots, k$ sollen zu einer Liste $A[1, \dots, kn]$ verschmolzen werden.

a) Betrachte den Algorithmus $\text{MERGE}(A, 1, n, 2n), \text{MERGE}(A, 1, 2n, 3n), \dots$

ges: Laufzeit

$\text{MERGE}(A, p, q, v)$ hat $O(v-p+1)$ ($v-p+1$ -mal.. Anzahl Elemente, die verschmolzen werden)

$\Rightarrow \text{MERGE}(A, 1, n, 2n)$ hat Laufzeit $2n$

$\text{MERGE}(A, 1, 2n, 3n)$ hat Laufzeit $3n$

\vdots
 $\text{MERGE}(A, 1, (j-1)n, jn)$ hat Laufzeit jn

\vdots
 $\text{MERGE}(A, 1, (k-1)n, kn)$ hat Laufzeit kn

\Rightarrow Algorithmus hat Laufzeit $\sum_{j=2}^k (jn) = n \cdot \frac{1}{2}(k^2 + k - 2) \Rightarrow O(nk^2)$



b) ges: Algorithmus mit besserer Laufzeit mittels Divide and Conquer

Algorithmus(A, l, r, n): // erster Aufruf mit Algorithmus($A, 1, k, n$)

if $(r-l) > 0$:

Algorithmus($A, l, \lfloor \frac{l+r}{2} \rfloor, n$);

Algorithmus($A, \lfloor \frac{l+r}{2} \rfloor + 1, r, n$);

Merge($A, (l-1)n+1, \lfloor \frac{l+r}{2} \rfloor \cdot n+1, r, n$);

$$T(z) = \begin{cases} \Theta(1) & \text{falls } z \leq 1 \\ 2 \cdot T(\frac{z}{2}) + n \cdot \Theta(z) & \text{falls } z > 1 \end{cases} \quad \left| \begin{array}{l} z = r-l = k-1 \approx k \end{array} \right.$$

$$\Rightarrow T(2^i) = n \cdot \Theta(2^i) + 2 T(2^{i-1})$$

$$= n \cdot \Theta(2^i) + 2n \cdot \Theta(2^{i-1}) + 2^2 T(2^{i-2})$$

$$= \sum_{j=0}^i 2^j \cdot n \cdot \Theta(2^{i-j}) = n \cdot \sum_{j=0}^i \Theta(2^i)$$

$$= n \cdot i \cdot \Theta(2^i) = n \cdot \ln_2(z) \cdot \Theta(z)$$

$$= n \cdot \ln(k) \cdot \Theta(k)$$

$$\Rightarrow O(n \cdot k \cdot \ln(k))$$

