

Sei $X := \{a_1, \dots, a_n\}$ die Menge der verschiedenen zu gruppierenden Objekte.

Wir definieren fuer $a_i, a_j \in X$ eine Aequivalenzrelation \sim so, dass $a_i \sim a_j$ genau dann gilt, wenn a_i und a_j die gleiche Markierung haben. Damit repraesentiert X/\sim die verschiedenen Gruppierungen.

Da die Anzahl der Markierungen $k \in \mathbb{N}$ konstant ist -das heisst insbesondere unabhaengig von n ist- ,finden wir eine Bijektion

$$\gamma : X/\sim \rightarrow \{1, \dots, k\}$$

Weiter definieren wir eine Funktion

$$\mu : X \rightarrow \{1, \dots, k\}, \mu(x) = \gamma(\pi(x))$$

, wobei π die kanonische Projektion

$$\pi : X \rightarrow X/\sim, \pi(x) = [x]_\sim$$

ist.

Wiederum liefert die Funktion

$$\alpha : \{1, \dots, k\} \rightarrow \mathbb{N}, \alpha(i) = |\gamma^{-1}(i)|$$

die Anzahl der Objekte in der i -ten Gruppe.

Wir setzen $B := (\beta_1, \dots, \beta_k)$, wobei $\beta_1 = 1$ und $\beta_j = 1 + \sum_{i=1}^{j-1} \alpha_i$ fuer alle $j = 2, \dots, k$.

Nun zum Algorithmus: Sein nun $x = (x_1, \dots, x_n)$ eine Folge mit Elementen aus X , wobei $x_i \neq x_j$ fuer alle $i \neq j$. Dieses x repraesentiert die ungruppierten Objekte. Man speichere das Element x_1 in einen Zwischenspeicher. Danach fuehre man folgende Anweisungen n mal aus:

1. Wir bezeichnen das jetzige Objekt im Zwischenspeicher mit y . Nun vertausche man y mit dem Element an der Stelle $\beta_{\mu(y)}$, sodass nun das Element $x_{\beta_{\mu(y)}}$ im Zwischenspeicher ist.
2. Man erhoehe den Eintrag $\beta_{\mu(y)}$ von B um 1.

Das war der Algorithmus.

Zur Berechnung von B benoetigen wir eine Laufzeit von $\mathcal{O}(n)$, indem wir x linear durchzaehlen. B belegt einen konstanten Speicher.

Der Algorithmus zum Gruppieren fuehrt n mal konstante Anweisungen durch und hat daher eine Laufzeit von $\mathcal{O}(n)$. Der Zwischenspeicher bedarf dabei auch nur einen konstanten Speicherplatz.