
Algorithmen auf Sequenzen

Abgabetermin: Samstag, den 1. Februar, 10⁰⁰ in Moodle

Aufgabe 1 (8 Punkte)

Betrachte das Wort $t = t_1 \cdots t_{17} = \text{ABANANAISANANANAS}$ sowie die zugehörige Burrows-Wheeler-Transformierte $\hat{t} = \hat{t}_1 \cdots \hat{t}_{18} = \text{\$NNBSNNNAAAAAAAAAI}$.

- a) Konstruiere die für die Rank-Select-Datenstruktur benötigten Arrays R , R' und R'' jeweils für B_A und B_N (für Anfragen vom Typ $\text{rank}_0(\cdot)$). Hierbei soll als Block-Länge $s' = 3$ und als Super-Block-Länge $s = 6$ verwendet werden (auch wenn dann $(s')^2 \neq s$ gilt).
- b) Beantworte die Rank-Anfragen $\text{rank}_0^{B_A}(17)$ und $\text{rank}_1^{B_N}(17)$ nach der in der Vorlesung vorgestellten Methode basierend auf den Arrays aus a).
- c) Konstruiere den Wavelet-Tree zu $\hat{t}$ für $t\$$.
- d) Bestimme die Werte von $\text{Occ}(A, 14)$ und $\text{Occ}(N, 14)$ nach der in der Vorlesung vorgestellten Methode aus dem Wavelet-Tree aus c).

Hinweis: Das zu betrachtende Alphabet ist $\Sigma \cup \{\$\} = \{\$, A, B, I, N, S\}$, wobei die Ordnung auf dem Alphabet durch die Reihenfolge gegeben ist.

Aufgabe 2 (6 Punkte)

Betrachte die unorientierte Permutation $\pi = (6, 7, 3, 4, 5, 8, 9, 1, 2)$.

- a) Wende den Algorithmus aus der Vorlesung zur 2-Approximation für die minimale Reversal-Distanz auf π an. Gib dabei alle Zwischenschritte an und erkläre, warum eine bestimmte Reversion angewendet wird (bzw. nicht angewendet wird).
- b) Zeichne den Breakpoint-Graphen $G(\pi)$ für π .
- c) Wende die scharfe untere Schranke für die benötigte Anzahl von Reversionen zum Sortieren mit Reversionen aus der Vorlesung auf π bzw. den Breakpoint-Graphen $G(\pi)$ aus b) an.

Tutoraufgabe 3 (Vorbereitung bis zum 30.01.25)

Gib eine optimale Folge von Reversionen zum Sortieren der Folge $(3, 4, 1, 2)$ an.