

Text Indexing and Information Retrieval

Übungsblatt 6

Besprechung: 29.11.2018

Aufgabe 1

„Implementieren“ Sie eine naive Scan-Lösung für RMQs (also ohne Vorverarbeitung) und vergleichen Sie deren Query-Zeit auf Arrays unterschiedlicher Größe und Beschaffenheit (z.B. zufällig, aufsteigend, etc.) mit den RMQ-Datenstrukturen aus der *succinct data structures library* (<https://github.com/simongog/sdsl-lite>). Ab welcher Array-Größe lohnt sich die Vorverarbeitung auf jeden Fall?

Aufgabe 2 (Theorie)

Zeigen Sie die komplette Datenstruktur für RMQs in $O(1)$ Zeit auf dem LCP-Array des Textes `alabararlabarbarla$`. Für dieses Beispiel setzen Sie die Blockgröße s fix auf 4 (statt auf $\frac{\log n}{4}$).

Aufgabe 3 (Theorie)

Die Vorberechnung aller RMQs innerhalb aller Blöcke der Länge s ist auch ohne “Lücken” möglich. Wie das geht, steht in dem Artikel https://doi.org/10.1007/11780441_5 in Abschnitt 3.1. Stellen Sie diese Lösung vor und zeigen Beispiele!

Aufgabe 4 (Theorie)

Zeigen Sie, wie man mit RMQs – evtl. mit zusätzlichen Datenstrukturen – die folgenden Anfragen in $O(1)$ Zeit beantworten kann:

- a) 1. *Range All-Equal Queries*: wahr, falls alle Elemente in $[\ell, r]$ gleich sind.
2. *Range Top-10 Queries*: gibt die zehn größten Elemente in $[\ell, r]$ aus.
3. *Range Less-x Queries*: wahr, falls alle Elemente in $[\ell, r]$ kleiner als x sind (x ist hierbei ein Query-Parameter).
- b) *Range MaxSum Queries*: welches Teilintervall $[u, v] \subseteq [\ell, r]$ maximiert die Teilsumme aller Teilintervalle von $[\ell, r]$? Gehen Sie hierbei davon aus, dass die Arrayeinträge Integers sind. **(Schwer!)**