

# Übung zu der Vorlesung Text-Indexierung (WS20/21)

Hinweise

Abgabe bis spätestens Montag, den 23.11.2020 um 12:00 Uhr in elektronischer Form.

Des Weiteren gelten alle organisatorischen Hinweise vom ersten Übungsblatt.

## Programmieraufgabe

(6 Punkte)

Die genaue Aufgabenstellung entnehmen Sie bitte der README.md des Aufgaben-Repositories. Um das Ihr Repository zu erzeugen folgen Sie bitte den folgenden Link (Link nur in der Moodle-internen Version). Die einzelnen Teilaufgaben geben hierbei die folgenden Punkte:

- (a) Erstellung eines Suffix-Arrays mit `std::sort` (1 Punkt)
- (b) Erstellung eines Suffix-Arrays mit Präfix-Doubling (5 Punkte)

Sie müssen sich bei Teilaufgabe (b) nicht auf den in der Vorlesung beschriebenen Präfix-Doubling-Algorithmus beschränken. Es geht bei dieser Aufgabe nur darum, dass Sie einen Suffix-Array-Konstruktionsalgorithmus umsetzen, der auf Präfix-Doubling basiert. Weitere Präfix-Doubling-Algorithmen finden Sie beispielsweise in folgendem Artikel beschrieben.

Roman Dementiev, Juha Kärkkäinen, Jens Mehnert, and Peter Sanders. “Better external memory suffix array construction”. In: *ACM J. Exp. Algorithmics* 12 (2008), 3.4:1–3.4:24. DOI: 10.1145/1227161.1402296

## Suche von Teilwörtern

(4 Punkte)

Für diese Aufgabe dürften Sie davon ausgehen, dass Sie Suffix-Arrays und LCP-Arrays in Linearzeit konstruieren können. Entwerfen Sie einen Linearzeitalgorithmus, der

- (a) für einen Text  $T$  das kürzeste Teilwort findet, das nur genau einmal in  $T$  vorkommt und (2 Punkte)
- (b) für zwei Texte  $T_1$  und  $T_2$  das längste Teilwort findet, welches sowohl in  $T_1$  als auch  $T_2$  vorkommt. (2 Punkte)

Bei mehreren Teilwörtern gleicher Länge können Sie ein beliebiges zurückgeben.