

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

Hinweise

Abgabe bis spätestens Montag, den 30.11.2020 um 12:00 Uhr in elektronischer Form.

Bitte geben Sie auch ein (leeres) PDF ab, wenn Sie nur die Programmieraufgabe lösen. Dies hilft bei der Verwaltung der Punkte in Moodle.

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) Klassifizierung der Suffixe in L, S und S^* (1 Punkt)
- (b) Sortierung der S^* -Substrings mithilfe von `std::sort` (0,5 Punkt)
- (c) Rekursiver Aufruf zum Sortieren der S^* -Suffixe (2,5 Punkte)
- (d) Induzieren aller Suffixe (2 Punkt)

Suffixsortierung

(4 Punkte)

Gegeben sei ein Text T der Länge n . Beweisen Sie, dass das Array A der Länge n ein Suffix-Array von T ist, genau dann wenn

1. A eine Permutation der Zahlen $[0, n)$ ist und
2. für alle i, j gilt $A^{-1}[i] \leq A^{-1}[j] \iff (T[i], A^{-1}[i+1]) \leq (T[j], A^{-1}[j+1])$.

Hierbei ist A^{-1} die inverse Permutation von A ist. Weiterhin können Sie annehmen, dass $A^{-1}[i] = 0$ für alle $i \geq n$.