

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

Hinweise

Abgabe bis spätestens Montag, den 25.01.2021 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

(4 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) Implementierung eines Bit-Vektors. (Als *alternative* Lösung, um keinen Bitvektor selber implementieren zu müssen, kann auch ein `std::vector<bool>` verwendet werden. Hierbei handelt es sich tatsächlich um einen Bitvektor. In diesem Fall gibt es aber nur 1 Punkt für die Aufgabe.) (2 Punkte)
- (b) Implementierung einer Rank-Datenstruktur. (3 Punkt)

Select-Anfragen

(6 Punkte)

Die zu *rank* inverse Operation ist *select*. So liefert $select_1(i)$ für $1 \leq i \leq n$ die Position der *i*-ten Eins in einem Bitvektor $B[1, n]$.

- (a) Zeigen Sie, wie sich select-Anfragen in $O(\lg n)$ Zeit mit einer Datenstrukturen der Größe $o(n)$ Bits beantworten lassen. (3 Punkte)
- (b) Nehmen Sie an, dass der Bitvektor B nur dünn besetzt ist, nämlich nur $o(\frac{n}{\lg n})$ Einsen hat. Geben Sie für diesen Fall eine Datenstruktur der Größe $o(n)$ Bits an, die select-Anfragen in konstanter Zeit beantworten kann. (3 Punkte)