Projekt für die Veranstaltung Text-Indexierung im WS 2023/24

Ziel des Projektes ist die Implementierung eines Suffix-Array-Konstruktionsalgorithmus. Bei diesem Algorithmus muss es sich nicht notwendigerweise um den in der Vorlesung besprochenen SAIS-Algorithmus handeln. Einzige Voraussetzung ist, dass es sich bei dem Algorithmus nicht um einen naiven Algorithmus handelt, d.h. der Algorithmus muss eine Laufzeit von $O(n \log n)$ oder $O(n)$ haben. Ein einfacher Aufruf einer Standardsortierfunktion mit einer entsprechenden Vergleichsfunktion reich *nicht*.

Neben dem Suffix-Array-Konstruktionsalgorithmus sollen zudem drei LCP-Array-Konstruktionsalgorithmen implementiert werden. Hierbei handelt es sich um den naiven Algorithmus, sowie die beiden in der Vorlesung vorgestellten Linearzeitalgorithmen.

Ein- und Ausgabe

Das Programm muss per Kommandozeile steuerbar sein. Die Eingabe hierfür hat das folgende Format.

```
ti_programm eingabe_datei
```

Bei der Eingabe-Datei handelt es sich immer um einen Text mit Byte-Alphabet. Ein einzelnes Zeichen ist somit immer ein Byte groß. Mögliche Beispieltexte sind im Pizza&Chili-Korpus (`http://pizzachili.dcc.uchile.cl/`) und im Manzini-Lightweight-Korpus (`https://people.unipmn.it/~manzini/lightweight/corpus/`) zu finden.

Die Ausgabe des Programms sieht wie folgt aus (Leerzeichen sind nur zwischen Parametern erlaubt):
`RESULT name=<your-name> sa_construction_time=<construction time (ms)>`
`sa_construction_memory=<memory peak (MiB)> lcp_naive_construction_time=<construction time (ms)>`
`lcp_kasai_construction_time=<construction time (ms)> lcp_phi_construction_time=<construction time (ms)>`

Minimalanforderungen

Das Projekt darf in einer beliebigen Programmiersprache umgesetzt werden. Wichtig ist aber, dass das Projekt auf einem Linux-System (Ubuntu 20.04.3 LTS) lauffähig ist! Dem Projekt sollte eine detaillierte Anleitung beiliegen, die beschreibt, was zum kompilieren/ausführen benötigt wird und wie genau das Programm ausgeführt werden kann. Das Programm muss das oben beschrieben Ein- und Ausgabeformat unterstützen.

Wichtig: **Es dürfen keine externen Bibliotheken verwendet werden**, die nicht von Florian Kurpicz genehmigt wurden. Bitte per Mail (`kurpicz@kit.edu`) nachfragen, bevor externe Bibliotheken eingebunden werden. Die Standardbibliothek der jeweiligen Programmiersprache kann allerdings ohne Fragen verwendet werden.

Dokumentation, Evaluation und Präsentation

Der Code muss so dokumentiert sein, dass dem Leser klar wird, was an welcher Stelle was genau passiert. Hierfür sollte darauf geachtet werden, dass die Dokumentation auch erklärt *warum* etwas gemacht wird und nicht nur *was* gemacht wird.

Die Evaluation ist Teil der Präsentation. Hier können unter anderem die Laufzeiten des eigenen Ansatzes für unterschiedliche Eingabe gezeigt werden. Ein Vergleich mit anderen Implementierungen ist auch möglich. (Hinweis: Das Ausgabeformat kann direkt zum Erstellen von Diagrammen genutzt werden, siehe `https://github.com/bingmann/sqlplot-tools/`.)

Die Ergebnisse sollen dann in einer ca. fünfminütigen Präsentation vorgestellt werden. In der Präsentation soll neben der Evaluation der Implementierung darauf eingegangen werden, was implementiert wurde, wie es implementiert wurde und was eventuell besonders an der Implementierung ist.

Wettbewerb

Des Weiteren gibt es noch einen kleinen Wettbewerb, an dem jedes eingereichte Projekt automatisch teilnimmt. Das Abschneiden im Wettbewerb hat keinen Einfluss auf die Note des Projekts! Bei dem Wettbewerb werden alle

Laufzeiten und der Speicherplatzbedarf gewichtet bepunktet. Die Gewichtung ist: 75 % Konstruktionszeit und 25 % Speicherplatz.

Deadline

Das Projekt muss bis zum 05.02.2024 um 23:59 Uhr deutscher Zeit per Mail an `kurpicz@kit.edu` gesendet werden (gerne als Link zu einem Repository). Spätere Abgaben können leider nicht berücksichtigt werden. Die Vorträge werden am 12.02.2024 stattfinden.