Projekt für die Veranstaltung Text-Indexierung im WS 2021/22

Ziel des Projektes ist die Implementierung eines (oder mehrerer) Textindizes, welche zur Beantwortung der folgenden beiden Anfragen genutzt werden können.

1. Welches ist das k. häufigst vorkommende Muster der Länge ℓ und
2. welches Muster ist das längste Muster der Form $\alpha\alpha$ (mit $\alpha \in \Sigma^*$)?

Wenn mehrere die Kriterien erfüllende Muster gleich häufig vorkommen, dann soll das lexikographisch kleinste zurückgegeben werden. Hierzu soll einer (mehrere) der, in der **Vorlesung Text-Indexierung vorgestellten**, Textindizes implementiert werden. Neben der Implementierung sind Dokumentation, Evaluation und Abschlusspräsentation ein Bestandteil des Projektes.

Ein- und Ausgabe

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

```
ti_programm [topk|repeat] eingabe_datei
```

Der Parameter *topk* sorgt dafür, dass Anfragen vom ersten Typ gestellt werden und der Parameter *repeat* sorgt dafür, dass Anfragen vom zweitem Typen gestellt werden. Die Eingabedatei hat je nach Parameter eine andere Struktur.

Top-k. Hier enthält die erste Zeile die Zahl $n \in \mathbb{N}_{>0}$ und wird gefolgt von n Zeilen, die aus zwei durch ein Leerzeichen getrennten nicht-negativen ganzzahligen Zahlen bestehen. Jede Zeile entspricht einer Anfrage. Die erst Zahl soll hierbei ℓ sein und die zweite Zahl k. Gefolgt werden diese Zeilen vom eigentlichen Text. Alle Anfragen werden eine Lösung haben, d.h., es wird nicht das achthäufigste Muster der Länge ℓ angefragt, wenn es dies nicht gibt. Beispiel:

```
3
4 3
5 1
2 2
Dies ist ein Beispiel
Beispiel Text.
```

Die Ausgabe hat hierbei das folgende Format zu haben:

```
RESULT algo=topk name=<your name> construction_time=<index construction time (ms)> query_time=<sum of
all query times (ms)> solutions=<semicolon separated list of solutions> file=<path to file>
```

Beispiel:

```
RESULT algo=topk name=florian construction_time=531 query_time=781 solutions=aa;abe;zzzz
file=/tmp/example_ti.txt
```

Repeat. Hier besteht die Eingabedatei ausschließlich aus dem Text. Beispiel:

```
Dies ist ein Beispiel
Beispiel Text.
```

Die Ausgabe hat hier ein ähnlich:

```
RESULT algo=repeat name=<your name> construction_time=<index construction time (ms)> query_time=<query
time (ms)> solution=<solution> file=<path to file>
```

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 der von mir durchgeführten Tests (gewichtet) addiert. Die geringste Laufzeit gewinnt. Die Gewichtung ist: 25 % Konstruktionszeit und 75 % Anfragezeit. Es lohnt sich also, die Anfragen zu optimieren.

Deadline

Das Projekt muss bis zum 31.01.2022 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.