Prof. Dr. Knut Reinert
Thimo Wellner
Sascha Meiers

Institut für Informatik
AG Algorithmische Bioinformatik

Algorithmen und Datenstrukturen in der Bioinformatik
Zweite Programmieraufgabe WS 13

Abgabe Montag, 18.11., 15:00 Uhr per SVN

In dieser Programmieraufgabe werden Sie die exakte Stringsuche mithilfe eines Suffixarrays implementieren.

Aufbau Schreiben Sie eine Funktion `void construct(vector<unsigned> & sa, string const & text)`, die ein leeres Suffixarray sowie den Text erhält und das Suffixarray konstruiert. Nutzen Sie die naive Konstruktionsmethode mit `std::sort`. Beachten Sie, dass ein Suffixarray niemals Strings speichert, sondern lediglich deren Startpositionen im Originaltext.

- Anders als in den theoretischen Aufgaben benötigen sie kein extra $ am Ende des Textes.
- Um `std::sort` zu nutzen müssen Sie einen sog. Funktor anlegen, der den *kleiner*-Operator für 2 Textpositionen definiert. Beispiele hierfür gibt es im Netz, u.A. unter `http://www.cplusplus.com/reference/algorithm/sort/`.

Suche Programmieren Sie die Binärsuche mit *mlr*-Heuristik in der Funktion `bool find(string const & query, vector<unsigned> const & sa, string const & text)`. Wie am Rückgabetyp erkennbar genügt es zu prüfen, **ob** das Suchwort im Text vorkommt oder nicht.

Aufruf Das Programm kann in 2 Modi ausgeführt werden. Gibt man nur ein Argument - den Text - an, so soll das Suffixarray zeilenweise ausgegeben werden. Gibt man nach dem Text noch eins oder mehrere Suchwörter als weitere Kommandozeilenparamter an, dann soll zeileweise für jedes Suchwort 1 oder 0 ausgegeben werden, je nachdem ob das Wort im Text enthalten ist oder nicht. Bei nicht korrektem Aufruf soll das Programm eine Fehlermeldung ausgeben und den return code 1 zurück geben.

Beispiele für beide Modi:

```
./aufgabe2 "banana"
5
3
1
0
4
2

./aufgabe2 "exact search using suffix arrays" search horspool "g suf"
1
0
1
```

Praktikumshinweise Beachten Sie die Hinweise unter `https://www.mi.fu-berlin.de/w/ABI/AlDaBiWS13Praktikum`.