Duarte Goncalves Mendes

Boyer–Moore String-search Algorithm

Der Boyer-Moore String-search Algorithmus ist dafür da, einen bestimmten Teiltext in einem Text zu suchen. Dies bewältigt er auf 2 verschiedenen Arten zusammen:

Bad-Character-Heuristik

Beim Bad-Character geht der Algorithmus zum Text und setzt denn Suchstring an Anfang an. Dannach sucht er den Ersten Buchstaben des Suchstrings von Rechts nach Links der nicht mit dem Text übereinstimmt. Hat es so einen dann ist der String nicht gleich und sommit verschiebt er denn String so weit nach Vorne das dieser Nicht Matchende (Bad Character) entweder nun im Suchstring an der Richtigen Stelle ist oder Falls dieser Buchstabe im Suchstring nicht vorkommt sodass der Suchstring direkt nach dem “Bad Character” beginnt. Dies tut er bis zum Schluss des Textes und hohlt sich somit alle gefundenen Positionen.

Good-Suffix-Heuristik

Der Good-Suffix funktioniert gleich wie der “Bad Character” jedoch mit dem Unterschied dass er bei einem Missmatch nicht einfach nach diesem Buchstaben sucht um das Wort möglichst weit zu verschieben sondern er sucht nach dem Längs möglichen Teilstring des Suchbegriffs (Sprich wo die Mehrheit der aufeinanderfolgenden Buchstaben im Suchstring und Text aufeinander passen, a.k.a. “Good Suffix”) und verschiebt ihn an diese Position. Gibt es keine aufeinanderfolgenden Buchstaben die Matchen so wird auf den Letzen Buchstaben gematched analog “Bad Character”.

Mittels diesen zwei Heuristiken berechnet der Algorithmus wie viel nach vorne Verschoben werden muss, um Möglichst wenige Iterationen zu brauchen und dennoch alle Vorkommnisse des Suchstrings im Text zu finden. Bei meinem Programm benutze ich jedoch die “Bad Character” auch wenn diese durch nur Verschieben und nicht “Springen” im Wort weniger Effizient ist da die “Good Suffix” variante sehr viel Komplexer zur Implementation ist.

Korrektheit

Zu der Korrektheit des Algorithmus habe ich nichts zu bemängeln, da er genau das Tut was er Tun soll und dies auch noch auf einer der Effizientesten arten (Wenn beide Heuristiken angewendet werden).

Effizienz

Die Effizienz des Boyer-Moore suchalgorithmus ist soweit ich Online fand einer der Effizientesten falls nicht sogar der Effizienteste String-such algorithmus. Dies bedingt jedoch dass er beide Heuristiken anwendet und somit relativ Komplex wird. Falls jedoch wie bei meiner kleinen Applikation nur den “Bad Character” anwendet dann sinkt die Effizienz um einiges obwohl es immernoch zu den Besten dann zählt.

Komplexität

Der Algorithmus ist recht Effizient auch bei längeren Texter da er über beide Heuristiken den schnellsten Weg bekommt und somit dann auch Schnell ist. Benutzt man Jedoch nur die Bad Character Heuristik dann wird er bei längeren Texten weniger Effizient da er immernoch Zeichen für Zeichen sozusagen durchprüft und überspringt anstatt ein paar Auszulassen wie bei der Good Suffix Heuristik.