

# Algorithmen und Datenstrukturen

## Übungsblatt 3

Ausgabe: 21. November — Besprechung: 2./4. Dezember

### Aufgabe 3.1 (Suffix-Arrays) [**Präsentation von Team: 9**]

Der Artikel

S. J. Puglisi, W. F. Smyth, and A. H. Turpin, *A Taxonomy of Suffix Array Construction Algorithms*, ACM Computing Surveys, 39(2), Article 4, June 2007

schlägt eine Klassifizierung von Algorithmen zur Konstruktion von Suffix-Arrays vor und evaluiert die unterschiedlichen Algorithmen experimentell anhand von Benchmark-Daten.

Präsentieren Sie diese Algorithmen-Klassen mit jeweils einem typischen Vertreter, sowie die Ergebnisse der experimentellen Studie. Verhalten sich alle Algorithmen einer Klasse in den Experimenten gleich? Gibt es Unterschiede bzw. Ausreißer und wie lassen sich diese erklären?

### Aufgabe 3.2 (Pizza & Chili) [**Präsentation von Team: 1, 6, 8**]

Die Webseite <http://pizzachili.dcc.uchile.cl> von Paolo Ferragina und Gonzalo Navarro enthält Algorithmen zur Konstruktion von Suffix-Arrays, die alle dieselbe API verwenden, sowie Testdaten aus verschiedenen Bereichen.

Wählen Sie aus den vorhandenen Algorithmen 2 bis 3 aus und untersuchen Sie diese experimentell anhand der Testdaten **DNA**, **ENGLISH** und **SOURCES**. Nehmen Sie von den Testdaten jeweils einen Präfix der Größe 10MB, um die Laufzeiten und den Platzverbrauch klein zu halten. Untersuchen Sie die Laufzeiten und den Platzverbrauch bei der Konstruktion des Suffix-Arrays, sowie die Suchzeiten (überlegen Sie sich sinnvolle Eingaben / Benchmarks für die Suche), und präsentieren Sie ihre Ergebnisse.

### Aufgabe 3.3 (String Matching) [**Präsentation von Team: 4, 7, 10**]

In der Vorlesung wurde darauf hingewiesen, dass man in der Praxis beim Boyer-Moore-Algorithmus häufig die `suffix[]`-Verschiebung einfach weglässt. Vergleichen Sie experimentell, wie sich diese Modifikation auf die Laufzeit des Algorithmus auswirkt. Überlegen Sie sich sinnvolle Benchmarks, für die Sie den Algorithmus mit und ohne `suffix[]`-Verschiebung aufrufen, und präsentieren Sie Ihre Ergebnisse.

Sie können den Algorithmus wahlweise selber implementieren oder öffentlich verfügbaren Code verwenden.