

Algorithmen und Datenstrukturen

Übungsblatt 2

Ausgabe: 6. November — Besprechung: 18./20. November

Aufgabe 2.1 (Amortisierte Analyse)

Analysieren Sie die Laufzeit von Union-Find mit gewichteter Vereinigungsregel und Pfadkomprimierung mit Hilfe der amortisierten Analyse. Genauer: Zeigen Sie, welche Laufzeit eine Folge von n MAKESET-Operationen, gefolgt von $n - 1$ UNION- und m FIND-Operationen hat.

Anmerkung: Diese Analyse findet sich in vielen Lehrbüchern zu Algorithmen und Datenstrukturen. Als Beispiel sei hier Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms*, MIT Press, 2001 genannt. Sie können natürlich auch andere Literatur zu Rate ziehen!

Aufgabe 2.2 (Priority-Queues)

Vergleichen Sie experimentell verschiedene Implementierungen von Priority-Queues (nicht notwendigerweise externe). Sie können dazu öffentlich verfügbare Implementierungen nutzen (<http://www.leekillough.com/heaps/#implem> enthält z.B. viele Verweise auf Implementierungen).

- Wählen Sie mindestens drei verschiedene Datenstrukturen aus (z.B. Fibonacci-Heaps, Binary-Heaps, HoT-Queues, R-Heaps usw.). Achten Sie dabei auch auf Vergleichbarkeit des Codes der verschiedenen Implementierungen!
- Überlegen Sie sich sinnvolle Benchmarks für die Evaluierung der Implementierungen.
- Vergleichen Sie die Implementierungen mit Hilfe Ihrer Benchmarks. Welche Schlüsse können Sie daraus ziehen?

Aufgabe 2.3 (String Matching)

Galil und Seifert haben einen String-Matching Algorithmus vorgestellt, der in-situ arbeitet. Präsentieren Sie, wie dieser Algorithmus funktioniert.

Originalliteratur: Z. Galil and J. Seiferas, *Time-space-optimal string matching*, Journal of Computer and System Sciences, 26(3), pp. 280–294, 1983.

Siehe auch: Problem 32-1 in: Cormen, Leiserson, Rivest, and Stein, *Introduction to Algorithms*, MIT Press, 2001.