

**Aufgabe 1** Dünn besetzte Polynome

10 Punkte

In der Vorlesung haben wir zwei Möglichkeiten gesehen, um Polynome von Grad  $n$  darzustellen. Die Koeffizientendarstellung wird ineffizient, wenn viele Koeffizienten des Polynoms 0 sind (so genannte *dünn besetzte Polynome*). Entwickeln Sie eine effiziente Darstellung für solche Polynome und beschreiben Sie Algorithmen zum Addieren, Multiplizieren und Auswerten in Ihrer Darstellung. Analysieren Sie die Effizienz Ihrer Algorithmen in Abhängigkeit von dem Grad  $n$  des Polynoms und von der Anzahl  $k$  der Koeffizienten, die nicht Null sind.

*Zusatzaufgabe (5 Punkte):* Implementieren Sie Ihre Datenstruktur in Java oder Python.

**Aufgabe 2** Experimentelles Sortieren

10 Punkte

Aus ALP2 kennen Sie diverse Sortierverfahren. Suchen Sie sich drei davon aus, und implementieren Sie diese in Java oder Python. Dabei soll ein Algorithmus mit worst-case Laufzeit  $\Theta(n^2)$  und ein Algorithmus mit worst-case Laufzeit  $\Theta(n \log n)$  vorkommen. Testen Sie Ihre Implementierung mit zufälligen Eingaben, und geben Sie außerdem für jeden Algorithmus eine Klasse von Eingaben an, die einen “Grenzfall” darstellt.

Messen Sie die Millisekunden, die Ihre Algorithmen für jeweils mindestens zehn verschiedene zufällige Eingaben für mindestens hundert verschiedene Eingabegrößen benötigen. Bereiten Sie die Ergebnisse graphisch auf.

Diskutieren Sie Ihre Erkenntnisse. Was sind die jeweiligen Vorteile der Algorithmen, was die Nachteile? Welchen Algorithmus würden Sie unter welchen Umständen verwenden? Was sagen die Ergebnisse über die O-Notation?

**Aufgabe 3** Die Türme von Hollywood

10 Punkte

Wahrscheinlich kennen Sie die Legende der Türme von Hanoi.

*Im großen Tempel von Benares steht eine Blechplatte, auf der drei vertikale Stangen aus Diamanten befestigt sind. An diesen Stangen sind  $n$  Scheiben aus massivem Gold angebracht, mit  $n$  verschiedenen Größen. Am Anfang der Zeit hat der Gott Brahma alle Scheiben auf einer Stange abgelegt, der Größe nach sortiert, mit der größten Scheibe ganz unten. Die Brahmanen im Tempel arbeiten unermüdlich und transportieren die Scheiben von Stange zu Stange, wobei sie jeweils nur eine Scheibe auf einmal verschieben und niemals eine größere Scheibe auf eine kleinere Scheibe legen dürfen. Wenn alle Scheiben auf der letzten Stange angekommen sind, beginnt der Weltuntergang.*

- (a) Beschreiben Sie einen rekursiven Algorithmus, mit dessen Hilfe die Brahmanen den Weltuntergang möglichst schnell herbeiführen können. Bestimmen Sie die benötigte Anzahl der Verschiebeoperationen genau. Wie viele Jahre müssen wir auf den Weltuntergang warten, wenn Brahma anfangs 100 Scheiben abgelegt hat und es einen Tag dauert, um eine Scheibe zu verschieben?
- (b) Kürzlich wurde der Tempel nach Kalifornien verlegt, wo die Brahmanen wesentlich lässiger zur Sache gehen. Mittlerweile sind die Scheiben nur noch aus vergoldetem Sperrholz, und die Stangen sind aus Plexiglas. Vor allem aber wurde die Anforderung an die Reihenfolge der Scheiben gelockert: die größte Scheibe an einer Stange muss immer noch ganz unten liegen, aber die Scheiben darüber können in beliebiger Reihenfolge kommen. Zum Schluss müssen die Scheiben aber natürlich wieder in der richtigen Ordnung auf der letzten Stange angekommen sein.

Beschreiben Sie einen effizienten Algorithmus für das gelockerte Problem. Wie viele Verschiebeoperationen braucht Ihr Algorithmus genau? Wie viele Jahre müssen wir jetzt auf den Weltuntergang warten, wenn wir mit 100 Scheiben anfangen und es einen Tag dauert, eine Scheibe zu verschieben?

Credits: Diese Aufgabe habe ich bei Jeff Erickson gefunden.

<http://www.cs.uiuc.edu/~jeffe/teaching/algorithms/hwex/s09/hw0.pdf>