

26. Stichwortsuche. Der kleinste Textausschnitt, 0 Punkte¹

Entwerfen und analysieren Sie einen effizienten Algorithmus für folgendes Problem:

Gegeben sind zwei sortierte Folgen $A = (a_1, a_2, \dots, a_m)$ und $B = (b_1, b_2, \dots, b_n)$. Jede Folge ist ein *Index*, der angibt, an welchen Stellen ein bestimmtes Wort in einem Text vorkommt. Gesucht ist das kleinste Intervall $[u, v]$, das sowohl Elemente von A als auch Elemente von B enthält.

Beispiel: Das Wort *teile* kommt an den Positionen $A = (11, 16, 42, 101, 125, 767)$ vor, *herrsche* an den Positionen $B = (2, 44, 289, 300)$. Dann ist $[42, 44]$ der kleinste Textausschnitt, der beide Wörter enthält.

27. (0 Punkte) Welches Wort steht an Position 43?
28. (0 Punkte) Lösen Sie die vorige Aufgabe für *drei* Eingabefolgen A, B, C .
29. Programmieraufgabe: Multiplikation langer Zahlen nach Karatsuba, 20 Punkte

Programmieren Sie die Multiplikation beliebig langer positiver Binärzahlen mit dem Karatsuba-Algorithmus in JAVA.

Sie können das Paket `java.math.BigInteger` zum Einlesen und Ausgeben (und zum Testen) verwenden, aber Sie dürfen natürlich nicht die Additions- und Multiplikationsroutinen dieses Pakets oder eines anderen Pakets verwenden. Sie können die rekursive Zerlegung fortführen wahlweise (a) bis auf einzelne Bits oder (b) bis zur einer Größe, wo man die Multiplikation mit den eingebauten `int`-Zahlen durchführen kann.

Ihr Programm soll vorzeichenlose Dezimalzahlen von der Standardeingabe einlesen (eine Zeile pro Zahl) und nach jeweils zwei gelesenen Zahlen das Produkt ausgeben.

Testen Sie Ihr Programm mit jeweils 10 zufälligen Zahlenpaaren der Längen $n = 1000, 2000, 3000, \dots, 10000$ Dezimalziffern, und ermitteln sie für jede Länge die mittlere Laufzeit in Millisekunden. Stellen Sie die Ergebnisse graphisch dar. Schätzen Sie empirisch die Laufzeit ab, indem Sie die Parameter C und α der Formel $T = Cn^\alpha$ an ihre experimentellen Daten anpassen.

Anleitung zur Abgabe der Programmieraufgabe: Laden Sie das ausführbare Programm als `jar`-Datei und die dokumentierten Quelltexte (zum Beispiel als `zip`-komprimierten Ordner) auf der KVV-Seite der Veranstaltung² hoch. Ihr Programm muss mit

```
> java -jar Multiplikation.jar
```

gestartet werden können. Fügen Sie eventuell eine `README`-Datei hinzu, wenn die Benutzung nicht selbsterklärend ist.

30. Master-Theorem im Fall (+) mit $f(n) = \Omega(n^{\gamma+\varepsilon})$?, 0 Punkte

Finden Sie asymptotische obere und untere Schranken die Lösung folgender Rekursion.

$$T(n) = 2T(\lceil n/2 \rceil) + f(n), \quad \text{mit } f(n) = \begin{cases} n^3, & \text{für } n \text{ gerade} \\ n^2, & \text{für } n \text{ ungerade} \end{cases}$$

Wie ist es, wenn man in der Definition von f die beiden Fälle vertauscht?

¹Abgewandelte Fassung einer Frage aus einem Einstellungsgespräch bei Google.

²<http://kvv.imp.fu-berlin.de>