Einzelprüfung „Theoretische Informatik / Algorithmen (vertieft)“

Einzelprüfungsnummer 66115 / 2012 / Herbst

Thema 1 / Aufgabe 4

(*maximale Teilsumme*)

Stichwörter: Teile-und-Herrsche (Divide-and-Conquer)

Gegeben ist ein Array a von ganzen Zahlen der Länge n, z. B. :

i	0	1	2	3	4	5	6	7	8	9
a_i	5	-6	4	2	-5	7	-2	-7	3	5

Im Beispiel ist also $n = 10$. Es soll die maximale Teilsumme berechnet werden, also der Wert des Ausdrucks

$$\max_{i,j \leq n} \sum_{k=1}^{j-1} a_k$$

Im Beispiel ist dieser Wert 8 und wird für $i = 8$,$j = 10$ erreicht. Entwerfen Sie ein Divide-And-Conquer Verfahren, welches diese Aufgabenstellung in Zeit $\mathcal{O}(n \log n)$ löst. Skizzieren Sie Ihre Lösung hinreichend detailliert.

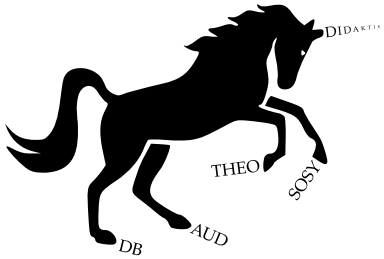
Tipp: Sie sollten ein geringfügig allgemeineres Problem lösen, welches neben der maximalen Teilsumme auch noch die beiden „maximalen Randsummen” berechnet. Die werden dann bei der Endausgabe verworfen.

```
/**
 * Klasse zur Berechnung der maximalen Teilsumme einer Zahlenfolge.
 *
 * nach Teilsumme.java Klasse mit Algorithmen für die Berechnung des größten
 * gemeinsamen Teilers zweier Ganzzahlen Algorithmen und Datenstrukturen,
 * Auflage 4, Kapitel 2.1
 *
 * nach Prof. Grude, Prof. Solymosi, (c) 2000-2008: 22. April 2008
 * <a href="http://public.beuth-hochschule.de/oo-
↪  plug/A&D/prog/kap21/Teilsumme.java">Teilsumme.java</a>
 */
public class Teilsumme {

  /**
   * Berechne die maximale Teilsumme an der rechten Grenze. Die Eingabeparameter
   * müssen diese Werte aufweisen: 0 &lt;= links &lt;= rechts &lt; folge.length.
   *
   * @param folge  Die Zahlenfolge, in der die maximale Zahlensumme gerechnet
   *               werden soll.
   * @param links  Die Index-Nummer der linken Grenze.
   * @param rechts Die Index-Nummer der rechten Grenze.
   *
   * @return Die maximale Teilsumme.
   */
  private static int berechneRandRechts(int[] folge, int links, int rechts) {
    int max = 0;
```

```
    int sum = 0;
    for (int i = rechts; i >= links; i--) {
      sum += folge[i];
      max = Math.max(max, sum);
    }
    return max;
  }

  /**
   * Berechne die maximale Teilsumme an der linken Grenze. Die Eingabeparameter
   * müssen diese Werte aufweisen: 0 &lt;= links &lt;= rechts &lt; folge.length.
   *
   * @param folge  Die Zahlenfolge, in der die maximale Zahlensumme gerechnet
   *               werden soll.
   * @param links  Die Index-Nummer der linken Grenze.
   * @param rechts Die Index-Nummer der rechten Grenze.
   *
   * @return Die maximale Teilsumme.
   */
  private static int berechneRandLinks(int[] folge, int links, int rechts) {
    int max = 0;
    int sum = 0;
    for (int i = links; i <= rechts; i++) {
      sum += folge[i];
      max = Math.max(max, sum);
    }
    return max;
  }

  /**
   * Berechne die maximale Teilsumme in der Zahlenfolge zwischen einer gegeben
   * linken und rechten Grenze. Die Eingabeparameter müssen diese Werte aufweisen:
   * 0 &lt;= links &lt;= rechts &lt; folge.length.
   *
   * @param folge  Die Zahlenfolge, in der die maximale Zahlensumme gerechnet
   *               werden soll.
   * @param links  Die Index-Nummer der linken Grenze.
   * @param rechts Die Index-Nummer der rechten Grenze.
   *
   * @return Die maximale Teilsumme.
   */
  private static int berechne(int[] folge, int links, int rechts) {
    if (links == rechts) // nur ein Element
      return Math.max(0, folge[links]);
    else {
      final int mitte = (rechts + links) / 2;
      final int maxLinks = berechne(folge, links, mitte);
      final int maxRechts = berechne(folge, mitte + 1, rechts);
      final int maxGrenzeRechts = berechneRandRechts(folge, links, mitte);
      // linke Hälfte
      final int maxGrenzeLinks = berechneRandLinks(folge, mitte + 1, rechts);
      // rechte Hälfte
      return Math.max(maxRechts, Math.max(maxLinks, maxGrenzeRechts + maxGrenzeLinks));
    }
  }
```

```java
  /**
   * Berechne die maximale Teilsumme einer Zahlenfolge rekursiv mit
   * logarithmischer Zeitkomplexität.
   *
   * @param folge Die Zahlenfolge, in der die maximale Zahlensumme gerechnet
   *              werden soll.
   *
   * @return Die maximale Teilsumme.
   */
  public static int berechne(int[] folge) {
    return berechne(folge, 0, folge.length - 1);
  }

  public static void main(String[] args) {
    int[] folge = { 5, -6, 4, 2, -5, 7, -2, -7, 3, 5 };
    int ergebnis = berechne(folge);
    System.out.println(ergebnis);
  }
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_66115/jahr_2012/herbst/Teilsumme.java`



Die Bschlangaul-Sammlung
Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an hermine.bschlangaul@gmx.net.Der TEX-Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: `https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Examen/66115/2012/09/Thema-1/Aufgabe-4.tex`