Einzelprüfung „Theoretische Informatik / Algorithmen / Datenstrukturen (nicht vertieft)“

Einzelprüfungsnummer 46115 / 2016 / Herbst

Thema 2 / Aufgabe 4

(*Methode „a()“*)

Stichwörter: Dynamische Programmierung

Mittels Dynamischer Programmierung (auch Memoization genannt) kann man insbesondere rekursive Lösungen auf Kosten des Speicherbedarf beschleunigen, indem man Zwischenergebnisse „abspeichert “ und bei (wiederkehrendem) Bedarf „abruft“, ohne sie erneut berechnen zu müssen.

Gegeben sei folgende geschachtelt-rekursive Funktion für $n, m \geq 0$:

$$a(n,m) = \begin{cases} n + \lfloor \frac{n}{2} \rfloor & \text{falls } m = 0 \\ a(1, m-1), & \text{falls } n = 0 \wedge m \neq 0 \\ a(n + \lfloor \sqrt{a(n-1,m)} \rfloor, m-1), & \textit{sonst} \end{cases}$$

(a) Implementieren Sie die obige Funktion `a(n,m)` zunächst ohne weitere Optimierungen als Prozedur/Methode in einer Programmiersprache Ihrer Wahl.

Lösungsvorschlag

```java
public static long a(int n, int m) {
  if (m == 0) {
    return n + (n / 2);
  } else if (n == 0 && m != 0) {
    return a(1, m - 1);
  } else {
    return a(n + ((int) Math.sqrt(a(n - 1, m))), m - 1);
  }
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46115/jahr_2016/herbst/DynamischeProgrammierung.java`

(b) Geben Sie nun eine DP-Implementierung der Funktion `a(n,m)` an, die `a(n,m)` für $0 \geq n \geq 100000$ und $0 \geq m \geq 25$ höchstens einmal gemäß obiger rekursiver Definition berechnet. Beachten Sie, dass Ihre Prozedur trotzdem auch weiterhin mit $n > 100000$ und $m > 25$ aufgerufen werden können soll.

Lösungsvorschlag

```java
static long[][] tmp = new long[100001][26];

public static long aDp(int n, int m) {
  if (n <= 100000 && m <= 25 && tmp[n][m] != -1) {
    return tmp[n][m];
  } else {
    long merker;
    if (m == 0) {
      merker = n + (n / 2);
    } else if (n == 0 && m != 0) {
      merker = aDp(1, m - 1);
```

```
      } else {
        merker = aDp(n + ((int) Math.sqrt(aDp(n - 1, m))), m - 1);
      }
      if (n <= 100000 && m <= 25) {
        tmp[n][m] = merker;
      }
      return merker;
    }
  }
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46115/jahr_2016/herbst/DynamischeProgrammierung.java`

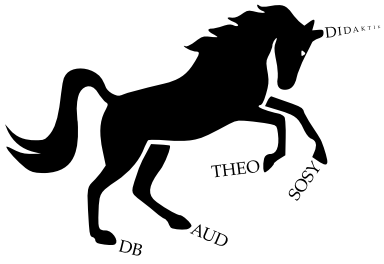
Lösungsvorschlag

Kompletter Code

```
public class DynamischeProgrammierung {
  public static long a(int n, int m) {
    if (m == 0) {
      return n + (n / 2);
    } else if (n == 0 && m != 0) {
      return a(1, m - 1);
    } else {
      return a(n + ((int) Math.sqrt(a(n - 1, m))), m - 1);
    }
  }

  static long[][] tmp = new long[100001][26];

  public static long aDp(int n, int m) {
    if (n <= 100000 && m <= 25 && tmp[n][m] != -1) {
      return tmp[n][m];
    } else {
      long merker;
      if (m == 0) {
        merker = n + (n / 2);
      } else if (n == 0 && m != 0) {
        merker = aDp(1, m - 1);
      } else {
        merker = aDp(n + ((int) Math.sqrt(aDp(n - 1, m))), m - 1);
      }
      if (n <= 100000 && m <= 25) {
        tmp[n][m] = merker;
      }
      return merker;
    }
  }

  public static void main(String[] args) {
    for (int i = 0; i < 100001; i++) {
      for (int j = 0; j < 26; j++) {
        tmp[i][j] = -1;
      }
    }
```

```
    System.out.println("schnell mit DP: " + aDp(7,7));
    System.out.println("langsam ohne DP: " + a(7,7));
  }
}
```

Code-Beispiel auf Github ansehen: `src/main/java/org/bschlangaul/examen/examen_46115/jahr_2016/herbst/DynamischeProgrammierung.java`



Die Bschlangaul-Sammlung
Hermine Bschlangauland 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 dieses Dokuments kann unter folgender URL aufgerufen werden: `https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Staatsexamen/46115/2016/09/Thema-2/Aufgabe-4.tex`