Power of Recursion Challenges

© 2023, Michael Inden

PART 1: Schnelleinstieg Rekursion

1) GCD-Berechnung

Schreiben Sie eine Methode int gcd(int, int), die den größten gemeinsamen Tei ler (ggT) berechnet (Im Englischen heißt dieser Greatest Common Divisor (GCD)). Dieser lässt sich mathematisch rekursiv wie folgt für zwei natürliche Zahlen a und b definieren:

$$gcd(a,b) = egin{cases} a, & b=0 \ gcd(b,a\,\%\,b), & b
eq 0 \end{cases}$$

Beispiele

Eingabe 1	Eingabe 2	Resultat
42	7	7
42	28	14
42	14	14

2) Array Sum

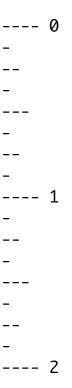
Schreiben Sie eine Methode int sum(int[]), die die Summe der Werte des übergebenen Arrays rekursiv errechnet.

Beispiele

Eingabe	Resultat
[1, 2, 3]	6
[1, 2, 3, -7]	-1

3) Ruler

In dieser Aufgabe wollen wir ein Lineal im englischen Stil nachahmen. Dabei wird ein Bereich von einem Inch in 1/2 und 1/4 sowie 1/8 unterteilt. Dabei nimmt die Länge der Striche jeweils um eins ab.



4) Palindrom

Schreiben Sie eine Methode boolean is Palindrome (String), die rekursiv überprüft, ob ein gegebener String unabhängig von Groß- und Kleinschreibung ein Palindrom ist. Ein Palindrom ist ein Wort, das sich von vorne und von hinten gleich liest.

Beispiele

Eingabe	Resultat
"Otto"	true
"ABCBX"	false
"ABCXcba"	true

5) BONUS: Berechne min + max mit möglichst wenigen Vergleichen – ASK ME as PO

PART 2: Suchen und Sortieren

1) Binary Search

Implementieren Sie die Binärsuche für ein int[] oder eine List<Integer>, die true oder false liefert, je nach Vorhandensein. Erweitere die Implementierung, so dass die Position gefunden wird.

2) Merge Sort

Schreiben Sie eine generische Methode static int[] mergesort(final int[] toSort), die ein Array mithilfe des Merge Sort Algorithmus sortiert und dabei Rekursion nutzt

PART 3: 2-D

PART 4: Fallstricke