ETH zürich
Departement Informatik

Programmieren und Problemlösen

Dr. Dennis Komm
https://lec.inf.ethz.ch/ppl/
17. Mai 2021

Übungsblatt 5

Aufgabe 1

Schreiben Sie ein Programm, welches per Divide-and-Conquer rekursiv das Maximum der Liste findet und zurückgibt.

Hinweis: Divide-and-Conquer ist eine Technik, bei der ein Problem in kleinere Teilprobleme zerlegt wird, die danach leichter zu lösen sind. Es ist beispielsweise einfach das Maximum einer Liste der Länge 1 zu bestimmen.

```
1 def maxi(daten):
2     # Ihr Code hier
```

Aufgabe 2

In der folgenden Implementierung von Quicksort gibt es einige Fehler. Versuchen Sie, diese zu finden und zu korrigieren.

```
1  def quicksort(daten):
2      if len(daten) > 1:
3          return daten
4      pivot = daten[0]
5      daten.pop(0)
6      left_part = [0 for l in daten if l < pivot]
7      right_part = [l for l in daten if l > pivot]
8      left_sorted = quicksort(right_part)
9      right_sorted = quicksort(left_part)
10     return left_sorted + pivot + right_sorted
11
12 a = [0, 0, 0, 0, 0, 0, 0, 0, 0]
13 b = [4, 1, 5, 2, 6, 7, 8, 9, 0]
14 c = [4, 5, 3, 7, 4, 6, 7, 4, 9]
15 d = []
16
17 print(quicksort(d))
```

Aufgabe 3

Vollziehen Sie die folgende Implementierung zur Berechnung der Fibonaccizahlen nach. In der Vorlesung haben Sie bereits zwei Varianten dieser Funktion gesehen, wobei die eine Mühe hatte, grosse Fibonaccizahlen zu berechnen. Erklären Sie, wieso die hier gezeigte Implementierung auch für `fibonacci(100)` gut funktioniert.

```
1 def fibonacci(n):
2     if n == 1:
3         return (1, 0)
4     fib1, fib2 = fibonacci(n-1)
5     return fib1 + fib2, fib1
6
7 for i in range(1,20):
8     a, b = fibonacci(i)
9     print(a)
```

Aufgabe 4

Schreiben Sie eine Funktion, die eine Liste als Parameter übergeben bekommt und daraus ein Anagramm erstellt. Verwenden Sie dazu Rekursion.

```
1 def list_mirror(daten):
2     # Ihr Code hier
3
4 print(list_mirror([1, 2, 3, 4, 5, 6, 7, 8, 9]))
```

Ausgabe:

```
1 [1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

Aufgabe 5

In dieser Aufgabe sollen Sie ein Programm schreiben, das ein Dictionary wie ein Telefonbuch verwendet.

Hinweis: Stellen Sie sicher, dass ein Wert im Telefonbuch vorhanden ist, bevor Sie ihn löschen.

```
1  def add(telefonbuch):
2      # Ihr Code hier
3
4  def delete(telefonbuch):
5      # Ihr Code hier
6
7  def drucken(telefonbuch):
8      # Ihr Code hier
9
10 telefonbuch = dict()
11 while True:
```

```
12      next = input("Add, Delete, Print oder Stop? ")
13      if next == "Stop":
14          break
15      elif next == "Add":
16          telefonbuch = add(telefonbuch)
17      elif next == "Print":
18          telefonbuch = drucken(telefonbuch)
19      elif next == "Delete":
20          delete(telefonbuch)
```