

## Aufgabensammlung

1. Die Fibonacci-Funktion ist wie folgt induktiv definiert:

$$(IA) \quad fib(0) =_{\text{def}} 0, fib(1) =_{\text{def}} 1$$

$$(IS) \quad fib(m) =_{\text{def}} fib(m-1) + fib(m-2) \text{ für } m \geq 2$$

Schreiben Sie eine nicht-rekursive PYTHON-Funktion `fib(m)` die für Argumente  $m \geq 0$  die Zahl  $fib(m)$  zurückliefert und während der Berechnung alle Vorgänger  $fib(0), \dots, fib(m-1)$  in dieser Reihenfolge mit **print** ausgibt.

2. Schreiben Sie eine PYTHON-Funktion `common_prefix(v,w)` die für zwei Zeichenketten das gemeinsame Präfix maximaler Länge zurückliefert. Für `v='abcdef'` und `w='ab12'` ist dies beispielsweise `'ab'`.

3. Deklarieren Sie eine PYTHON-Funktion `two_in_three(A,B,C)` die die Menge aller Elemente zurückliefert, die in genau zwei der drei angegebenen Mengen `A, B, C` enthalten sind.

4. Schreiben Sie einen naiven Primzahltest  $\text{prime}(m)$  der für ganze Zahlen  $m \geq 2$  durch Prüfen der Teiler  $2, \dots, \sqrt{m}$  feststellt, ob  $m$  eine Primzahl ist oder nicht. Ist 3203431780337 eine Primzahl?

5. Deklarieren Sie:

- (a) Die Funktion `word_length(words)` liefert für eine gegebene Liste `words` von Zeichenketten eine Liste der Längen der Wörter in `words` zurück.
- (b) Die Funktion `squares(x, y)` liefert für ganze Zahlen  $0 \leq x \leq y$  eine Liste aller Quadratzahlen zwischen  $x$  und  $y$ , ggf. einschließlich dieser Grenzen.

6. Schreiben Sie eine Funktion `add_number (contacts, name, number)`, wobei gilt:

- `contacts` ist eine assoziative Liste mit Einträgen vom Typ **`str:list`**,
- `name` ist von Typ **`str`**, und
- `number` ist von Typ **`int`**.

Ihre Funktion fügt `number` der Liste `contacts[name]` hinzu und vermeidet dabei doppelte Einträge. Gibt es noch keine solche Liste in `contacts`, wird sie neu angelegt.

7. Deklarieren Sie eine Funktion  $d(p, q)$ , die die *Euklidische Distanz* zweier Punkte  $p, q$  berechnet. Dabei sind die Punkte vom Typ **tuple** mit jeweils zwei ganzzahligen Koordinaten.

8. Schreiben Sie eine Funktion `closest_pair(U)`, die für eine Punktmenge  $U$  das Punktepaar mit kleinster Euklidische Distanz und deren Abstand zurückliefert (siehe vorhergehende Aufgabe). Sie können davon ausgehen, dass  $U$  mindestens zwei Punkte enthält.



9. Deklarieren Sie eine Funktion `edgeset_di(G)`, die für einen Digraphen mit Kantenkosten die Summe aller Kantenkosten mit **print** ausgibt und die Menge aller Kanten zurückliefert.

10. Schreiben Sie eine Funktion `valid_brackets(w)`, die für eine Zeichenkette `w` mit Hilfe eines Stacks feststellt, ob ggf. vorhandene Klammern `' ) '` und `' ( '` richtig gesetzt sind. Dazu wird jede öffnende Klammer auf den Stack gelegt und mit jeder schließenden Klammer ein Element vom Stack gelöscht. Was muss zum Schluss gelten?

11. Geben Sie jeweils *generator expressions* an:

- (a) Die maximale Wortlänge eines Wortes innerhalb einer Zeichenkette (Worte sind durch Leerzeichen getrennt). Recherchieren Sie die String-Methode `str.split()`.
- (b) Das Skalarprodukt zweier gleichlanger Vektoren, gegeben als Tupel.

12. Deklarieren Sie folgende Mengen:

- (a) Die Menge  $S$  aller Teilmengen von  $\{0, \dots, 9\}$  bei denen die Summe aller Elemente gerade ist.
- (b)  $T = \{2 \leq p < 1000 \mid p \text{ prim} \}$

13. Schreiben Sie eine Funktion `all_monotonic_functions(m, k)`, die für  $m, k \geq 1$  durch Iteration die Anzahl derjenigen Funktionen unter allen endlichen totalen Funktionen  $f : \{0, \dots, m-1\} \mapsto \{0, \dots, k-1\}$  bestimmt, die monoton wachsend sind. Jede solche Funktion soll über **print** ausgegeben werden.

14. Messen Sie die Laufzeit von `closest_pair(U)` für zufällige Punktmengen mit  $100, 200, \dots, 1000$  Elementen, vgl. Aufgabe 8.

15. Testen Sie Ihre Implementierung von `valid_brackets(w)` (vgl. Aufgabe 10) per Unittest für einige Zeichenketten

- mit mehr ( als ),
- mit gleicher Anzahl ( und ), aber falscher Reihenfolge,
- mit mehr ) als (, sowie
- mit korrekten Klammerschachtelungen.

16. (\*) Eine Palindromzahl besteht vorwärts und rückwärts gelesen aus der gleichen Ziffernfolge. Die größte Palindromzahl, die aus dem Produkt zweier zweistelliger Zahlen gebildet werden kann, ist  $9009 = 91 \cdot 99$ . Was ist die größte Palindromzahl, die aus dem Produkt zweier dreistelliger Zahlen gebildet werden kann?



17. (\*) Für  $n > 1$  entsteht die Collatz-Folge durch wiederholte Anwendung folgender Regeln:

- Falls  $n$  gerade, dann halbiere.
- Falls  $n$  ungerade, setze mit  $3n + 1$  fort.

Für den Startwert  $n = 13$  ergibt sich beispielsweise die Folge

$$13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

der Länge 10. Für welchen Startwert kleiner einer Million entsteht die längste Folge und wie lang ist sie?

18. (\*) Die Zahl 342 wird auf Englisch in Worten als 'three hundred and forty-two' geschrieben und umfasst 23 Buchstaben (ohne Leer- und Sonderzeichen). Für 115 sind es wegen 'one hundred and fifteen' genau 20 Buchstaben. Wenn alle Zahlen zwischen 1 und 1000 ausgeschrieben werden, wieviele Buchstaben werden dabei insgesamt verwendet?

19. (\*) Beginnend mit 1 und dann nach rechts im Uhrzeigersinn, entsteht folgende spiralförmige  $5 \times 5$ -Anordnung von Zahlen:

21	22	23	24	25
20	7	8	9	10
19	6	1	2	11
18	5	4	3	12
17	16	15	14	13

Die Summe der Diagonalelemente (hervorgehoben) ist 101. Wie lautet die Summe der Diagonalelemente in einer  $1001 \times 1001$ -Anordnung?

Die mit (\*) gekennzeichneten Aufgaben stammen aus dem Projekt Euler, siehe <http://projecteuler.net>.