Řešení domácích úkolů - 4. týden

Fibonacciův algoritmus pro rozklad zlomků

Mějme dvě celá čísla x, y, a rozložme zlomek $\frac{x}{y}$ na součet zlomků s čitatelem 1,

$$\frac{x}{y} = \frac{1}{c_1} + \frac{1}{c_2} + \cdots + \frac{1}{c_n} \tag{1}$$

Na rozklad použijeme Fibonacciův algoritmus ze 13. století, založený na rovnosti

$$\frac{x}{y} = \frac{1}{\lceil \frac{y}{x} \rceil} + \frac{-y \mod (x)}{y \lceil \frac{y}{x} \rceil} \tag{2}$$

kde $\lceil x \rceil$ je funkce "strop", tedy nejmenší celé číslo větší nebo rovné x; v Pythonu tuto funkci najdete jako `math.ceil()`. Druhou část vztahu budeme rozkládat podle stejného vztahu, dokud nedostaneme čitatel 1 i u druhého zlomku.

Nechť je například $x = 17, y = 36$. Pak

$$\frac{17}{36} = \frac{1}{\lceil \frac{36}{17} \rceil} + \frac{-36 \mod (17)}{36 \lceil \frac{36}{17} \rceil} = \frac{1}{3} + \frac{15}{36 \cdot 3} = \frac{1}{3} + \frac{5}{36} \tag{3}$$

A když použijeme stejný postup na zlomek vpravo:

$$\frac{17}{36} = \frac{1}{3} + \frac{1}{\lceil \frac{36}{5} \rceil} + \frac{-36 \mod (5)}{36 \lceil \frac{36}{5} \rceil} = \frac{1}{3} + \frac{1}{8} + \frac{4}{36 \cdot 8} = \frac{1}{3} + \frac{1}{8} + \frac{1}{72} \tag{4}$$

Implementujte tento algoritmus v Pythonu. Na vstupu dostanete dvě celá čísla, oddělené mezerou, například `17 36`. Na výstup napíšete seznam jmenovatelů rozkladu, oddělených mezerou a seznam ukončíte znakem nového řádku, tedy například `3 8 72\n`.

Příklad 1

Vstup:

1 2

Výstup:

2

Příklad 2

Vstup:

2 3

Výstup:

2 6

Příklad 3

Vstup:

21 32

Výstup:

2 7 75 16800

Poznámky

1. Testy k této úloze jsou všechny příklady, uvedené v zadání úlohy.
2. Svůj kód lehko otestujete převodem zlomků na desetinná čísla a porovnáním původního zlomku s jeho rozkladem.

Řešení

Analýza Tady stačí jednoduše realizovat rekurzivní algoritmus, dokud zlomek nezredukujeme do požadovaného tvaru.

Vzorové řešení

Toto řešení pouze věrně implementuje algoritmus.

```
from math import ceil, gcd

x, y = [int(s) for s in input().split()]
```

```
denoms = []

while x > 0:
    denom = ceil(y / x)
    denoms.append(denom)
    x, y = -y % x, y * denom
    divisor = gcd(x, y)          # Zkrátíme čitatele a j
    x, y = x // divisor, y // divisor

print(*denoms)
```

Alternativní řešení

Protože každý jmenovatel egyptské reprezentace dopočítáváme jako poměr x a y, nemusíme x a y krátit. Dostaneme tak poněkud kompaktnější kód:

```
from math import ceil

x, y = [int(s) for s in input().split()]

denoms = []

while x > 0:
    denoms.append(ceil(y / x))
    x, y = -y % x, y * denoms[-1]

print(*denoms)
```

Python má modul `fractions`, který implementuje objekt zlomku - `Fraction` - a operace se zlomky.

```
from math import ceil
from fractions import Fraction

x, y = [int(s) for s in input().split()]

f = Fraction(x, y)

denoms = []

while f.numerator != 1:
```

```
    denoms.append(ceil(f.denominator / f.numerator))
    f = Fraction(-f.denominator % f.numerator, f.denominator *
denoms[-1])

denoms.append(f.denominator)

print(*denomxs)

# Kontrola:
# print(Fraction(x, y) == sum([Fraction(1, d) for d in denoms]))
```

Mimochodem, tady dochází k tichému zkrácení čitatele a jmenovatele - zabezpečuje to objekt Fraction.