Lista 4

Uwaga (do Zadań 3 i 4): `len()` działa też dla list.

Zadanie 1 (1 punkt). *Liczby Leonarda* to następująco zdefiniowany ciąg liczbowy:

$$a_0 = 1,\ a_1 = 1,$$
$$a_n = a_{n-1} + a_{n-2} + 1\ (n \geqslant 2).$$

Napisz funkcje obliczające wartość a_n trzema sposobami: iteracyjnie, rekurencyjnie, oraz z wykorzystaniem wzoru[1]:

$$a_n = 2 \cdot \left\lfloor \frac{\phi^{n+1}}{\sqrt{5}} + \frac{1}{2} \right\rfloor - 1,\ \text{gdzie } \phi = \frac{1+\sqrt{5}}{2}.$$

Ze względu na sposób, w jaki Python reprezentuje liczby rzeczywiste, implementacja za pomocą wzoru może nie zwracać poprawnych wyników. Znajdź najmniejsze n, dla którego wersja używająca wzoru podaje wynik różniący się od rzeczywistego.

Zadanie 2 (1 punkt). Użyj funkcji `gcd` z wykładu (lub z modułu `math`) do napisania funkcji `simplify(p,q)`, która dla podanych liczb całkowitych `p` i `q` zwraca parę (krotkę) liczb całkowitych p', q' taką, że:

- $p'/q' =$ `p/q`,
- p'/q' jest ułamkiem nieskracalnym,
- q' jest dodatnia.

Możesz założyć, że parametr `q` jest różny od 0.

Zadanie 3 (1 punkt). Wektory w przestrzeni $\mathbb{R}^n$ można reprezentować jako listy liczb rzeczywistych długości n. Napisz funkcję `dot(w,v)`, która dla list `w` i `v` reprezentujących wektory równej długości zwraca ich iloczyn skalarny.

Zadanie 4 (1 punkt). Permutacją zbioru X nazywamy dowolną bijekcję $f : X \to X$. Permutację σ zbioru $X_n = \{0, 1, \ldots, n-1\}$ można reprezentować jako listę `lst` długości n taką, że `lst[i]` $= \sigma($`i`$)$ dla wszystkich $0 \leqslant$ `i` $< n$.

(1) (0,5 punktu) Napisz funkcję `inverse(sigma)`, która dla listy `sigma` długości n reprezentującej permutację σ zbioru X_n zwraca listę reprezentującą permutację σ^{-1}.

(2) (0,5 punktu) Napisz funkcję `composition(sigma1, sigma2)`, która dla list `sigma1`, `sigma2` długości n reprezentujących odpowiednio permutacje σ_1, σ_2 zbioru X_n zwróci listę reprezentującą permutację $\sigma_1 \circ \sigma_2$.

Wskazówka. Możesz użyć następującej konstrukcji: dla liczby naturalnej (lub wyrażenia) `n`, wyrażenie `list(range(n))` tworzy listę `[0, 1, 2, ..., n - 1]` długości `n`.

[1] $\lfloor x \rfloor$, czyli *podłoga z x*: zaokrąglenie x w dół.