

# Laboratorium kryptograficzne dla licealistów 4

## Rozwiązania zadań

Projekt „Matematyka dla ciekawych świata”

Łukasz Mazurek

20.04.2017

**Zadanie 1** Napisz funkcję `odl(f, g)`, która obliczy odległość pomiędzy rozkładami  $f$  i  $g$ . Funkcja powinna przyjmować jako argumenty dwie listy  $f$  i  $g$  tej samej długości. Funkcja powinna zwracać jako wynik sumę wartości bezwzględnych różnic pomiędzy listami na poszczególnych pozycjach:

$$|f(0) - g(0)| + |f(1) - g(1)| + \dots$$

Do obliczania wartości bezwzględnej liczby możesz użyć wbudowanej funkcji Pythona `abs()`.

Używając napisanej funkcji, oblicz odległość pomiędzy rozkładami z rys. 1a dla pierwszych pięciu liter alfabetu. Wywołanie funkcji powinno wyglądać następująco:

```
f = [0.093, 0.013, 0.018, 0.040, 0.005]
g = [0.085, 0.021, 0.037, 0.017, 0.037]
print(odl(f, g))
```

```
0.09
```

Rozwiązanie:

```
def odl(f, g):
    suma = 0
    for i in range(len(f)):
        suma = suma + abs(f[i] - g[i])
    return suma
```

**Zadanie 2** Napisz program, który w pętli obliczy wszystkie 32 rotacje listy `ile_tekst1` i dla każdej rotacji wypisze linijkę zawierającą numer tej rotacji i po dwukropku odległość tej rotacji od listy `ile_lalka`:

```
0: 0.826033617741158
1: 0.8483552668940029
2: 0.8266492142552446
...
```

Przypomnienie:  $k$ -tą rotacją listy nazywamy jej cykliczne przesunięcie w lewo o  $k$  pozycji. Np. drugą rotacją listy  $[1, 2, 3, 4, 5, 6]$  jest lista  $[3, 4, 5, 6, 1, 2]$ . Zatem  $k$ -tą rotację listy możemy uzyskać poprzez sklejenie fragmentu listy od  $k$ -tej pozycji (włącznie) do końca oraz fragmentu listy od początku do  $k$ -tej pozycji (wyłącznie).

Rozwiązanie:

```
for k in range(32):
    print(k, end = ' ')
    print(':', end = ' ')
    print(odl(ile_lalka, ile_tekst1[k:] + ile_tekst1[:k]))
```

**Zadanie 3** Napisz funkcję `znajdz_przesuniecie(f, g)`, która zwróci przesunięcie rozkładu  $g$  względem rozkładu  $f$ , czyli takie  $k$ , że  $k$ -ta rotacja rozkładu  $g$  jest najbliżej rozkładu  $f$ .

Funkcja powinna używać dwóch zmiennych: `min_odleglosc` oraz `najlepsze_przesuniecie`. W każdym kroku zmienna `min_odleglosc` powinna przechowywać najmniejszą odległość, jaką udało nam się do tej pory uzyskać, a zmienna `najlepsze_przesuniecie` powinna przechowywać numer rotacji, dla której ta najmniejsza odległość wystąpiła. Funkcja powinna w pętli sprawdzać kolejne przesunięcia  $k$  i w przypadku, gdy dane przesunięcie daje mniejszą odległość niż `min_odleglosc`, aktualizować obie zmienne.

Użyj funkcji `znajdz_przesuniecie`, aby znaleźć przesunięcie rozkładu `ile_tekst1` względem rozkładu `ile_lalka`:

```
print(znajdz_przesuniecie(ile_lalka, ile_tekst1))
```

21

Rozwiązanie:

```
def znajdz_przesuniecie(f, g):
    min_odleglosc = odl(f, g)
    najlepsze_przesuniecie = odl(f, g)
    for k in range(32):
        odleglosc = odl(f, g[k:] + g[:k])
        if (odleglosc < min_odleglosc):
            najlepsze_przesuniecie = k
            min_odleglosc = odleglosc
    return najlepsze_przesuniecie
```

**Zadanie 4** Używając powyższej funkcji `zlicz` oraz napisanych wcześniej funkcji, oblicz rozkład częstości liter w co piątej literze tekstu z pliku `tekst2.txt`<sup>1</sup>. Następnie oblicz najlepsze przesunięcie tego rozkładu względem rozkładu wszystkich liter z pliku `lalka.txt`. Na koniec oblicz odległość otrzymanego rozkładu przesuniętego o to najlepsze przesunięcie od rozkładu z Lalki.

Wskazówka: aby obliczyć oba rozkłady, wczytaj oba pliki, a następnie wywołaj funkcję `zlicz` z następującymi argumentami: `zlicz(lalka, 0, 1)` oraz `zlicz(tekst2, 0, 5)`. Otrzymane rozkłady zapisz w zmiennych `ile_lalka` oraz `ile_tekst2`. Następnie użyj funkcji `znajdz_przesuniecie`, aby znaleźć najlepsze przesunięcie rozkładu `ile_tekst2` względem rozkładu `ile_lalka`. Na koniec użyj funkcji `odl`, aby obliczyć odległość między odpowiednio przesuniętym rozkładem `ile_tekst2`, a rozkładem `ile_lalka`.

Twój program powinien wypisać komunikat takiej postaci:

Najlepsze przesuniecie:

21

Odleglosc:

0.07712184832382302

Rozwiązanie:

```
alfabet = 'AĄBCĆDEĘFGHIJKLŁMNŃOÓPRSŚTUWYZŻŻ'

def zlicz(tekst, poczatek, krok):
    ile = [0] * 32
    suma = 0
    licznik = 0
    for znak in tekst:
        if licznik % krok == poczatek:
            if znak in alfabet:
                poz = alfabet.index(znak)
```

<sup>1</sup>Plik `tekst2.txt` również należy pobrać ze strony [ciekawi.icm.edu.pl](http://ciekawi.icm.edu.pl)

```

        ile[poz] = ile[poz] + 1
        suma = suma + 1
    licznik = licznik + 1
return [x / suma for x in ile]

def odl(f, g):
    suma = 0
    for i in range(len(f)):
        suma = suma + abs(f[i] - g[i])
    return suma

def znajdz_przesuniecie(f, g):
    min_odleglosc = odl(f, g)
    najlepsze_przesuniecie = odl(f, g)
    for k in range(32):
        odleglosc = odl(f, g[k:] + g[:k])
        if (odleglosc < min_odleglosc):
            najlepsze_przesuniecie = k
            min_odleglosc = odleglosc
    return najlepsze_przesuniecie

plik = open('lalka.txt', 'r')
lalka = plik.read().upper()
plik.close()
ile_lalka = zlicz(lalka, 0, 1)

plik = open('tekst2.txt', 'r')
tekst2 = plik.read().upper()
plik.close()
ile_tekst2 = zlicz(tekst2, 0, 5)

przes = znajdz_przesuniecie(ile_lalka, ile_tekst2)
d = odl(ile_lalka, ile_tekst2[przes:] + ile_tekst2[:przes])

print('Najlepsze przesuniecie:')
print(przes)
print('Odleglosc:')
print(d)

```