

Laboratorium kryptograficzne dla licealistów 3

Projekt „Matematyka dla ciekawych świata”

Łukasz Mazurek

06.04.2017

1 Wczytywanie danych z pliku

Do tej pory wszystkie dane, z których korzystały nasze programy, wprowadzaliśmy bezpośrednio do kodu programu. Tym razem spróbujemy rozszyfrować dłuższą wiadomość zapisaną w pliku.

1.1 Pierwszy program — powiększacz liter

Napišemy program, który wczytuje tekst z pliku, powiększa go (zamienia wszystkie małe litery na wielkie) i zapisuje w innym pliku. Do tego zadania będziemy potrzebowali oddzielnego pliku z tekstem do obróbki. W tym celu otwórz stronę `repl.it` i kliknij przycisk *Add new file* w lewym górnym rogu strony. Pojawi się informacja o przejściu do trybu *Project Mode* i aktualnie edytowanemu plikowi zostanie nadana nazwa `main.py`. Utwórz nowy plik, klikając ponownie przycisk *Add new file* i nadaj nowo powstałemu plikowi nazwę `tekst1.txt`. Następnie otwórz plik `tekst1.txt` ze strony `ciekawicm.edu.pl` (zakładka *Dla uczestników* → *Materiały - ćwiczenia* → *Lab 3.*) i skopiuj całą jego treść do naszego pliku `tekst1.txt` na stronie `repl.it`.

Teraz zmień zawartość pliku `main.py` na następującą:

```
plik = open('tekst1.txt', 'r')
tekst1 = plik.read()
plik.close()
print(tekst1)
```

i uruchom program przyciskiem *run*. Program powinien wypisać zawartość pliku `tekst1.txt` do prawej konsoli. Co się stało?

- Polecenie z pierwszej linijki otwiera plik `tekst1.txt` i zapewnia dostęp do niego poprzez zmienną `plik`. Opcja `'r'` oznacza, że plik jest otwarty „do odczytu” (od angielskiego *read*).
- Druga komenda czyta zawartość pliku `tekst1.txt` i zapisuje go w zmiennej `tekst1`.
- Trzecia komenda zamyka dostęp do pliku `tekst1.txt`.
- Ostatnie polecenie wypisuje na ekran zawartość zmiennej `tekst1`.

Do zamiany liter małych na wielkie w Pythonie służy polecenie `upper()`. Aby w tekście zapisanym w zmiennej `napis` zmienić wszystkie litery małe na wielkie (a wielkie litery i pozostałe znaki pozostać niezmienione) należy użyć konstrukcji `napis.upper()`. Zmodyfikujmy zatem nasz program tak, żeby zmienna `tekst1` zawierała zawartość pliku `tekst1.txt` powiększoną przy użyciu polecenia `upper()`:

```
plik = open('tekst1.txt', 'r')
tekst1 = plik.read().upper()
plik.close()
print(tekst1)
```

Tym razem program powinien wypisać na ekran tekst składający się wyłącznie z wielkich liter i znaków niebędących literami.

Chcielibyśmy, aby tekst wygenerowany przez nasz program nie wyświetlał się w konsoli, tylko był zapisywany do nowego pliku `tekst1BIG.txt`. W tym celu musimy utworzyć nowy plik przy użyciu `open` z parametrem `'w'` (od angielskiego *write*) i użyć polecenia `write`. Ostatecznie nasz program powinien wyglądać następująco:

```
plik = open('tekst1.txt', 'r')
tekst1 = plik.read().upper()
plik.close()
plik = open('tekst1BIG.txt', 'w')
plik.write(tekst1)
plik.close()
```

Po wykonaniu powyższego programu na liście otwartych plików powinien pojawić się nowy plik o nazwie `tekst1BIG.txt` zawierający tekst z pliku `tekst1.txt` z małymi literami zamienionymi na wielkie.

1.2 Zliczanie wystąpień liter w tekście

Napiszemy teraz funkcję `zlicz(tekst)`, która zlicza liczbę liter A występujących w pliku `tekst1.txt`. Ponieważ wielkość liter chwilowo nie ma dla nas znaczenia, w tym zadaniu (jak i w kolejnych dotyczących zliczania liter) zawsze będziemy najpierw „powiększać” wszystkie litery wczytywanego pliku.

Stwórz program o następującej treści:

```
def zlicz(tekst):
    ile = 0
    for znak in tekst:
        if znak == 'A':
            ile = ile + 1
    return ile

plik = open('tekst1.txt', 'r')
tekst1 = plik.read().upper()
plik.close()
ile = zlicz(tekst1)
print(ile)
```

Powyższy kod składa się z definicji funkcji `zlicz(tekst)` oraz z czterech linii uruchamiających tę funkcję na tekście z pliku `tekst1.txt`¹. W funkcji `zlicz(tekst)`:

- Zmienna `ile` (początkowo równa 0) przechowuje aktualną liczbę zliczonych liter A.
- Za każdym razem, gdy trafimy w tekście na kolejną literę A, zwiększamy wartość zmiennej `ile` o jeden.
- Na koniec *zwracamy* wartość zmiennej `ile` jako wynik funkcji `zlicz(tekst)` przy użyciu instrukcji `return ile`.

W dalszej części kodu:

- Wczytujemy całą zawartość pliku `tekst1.txt`, powiększamy i zapisujemy w zmiennej `tekst1`.

¹Zwróć uwagę, że w powyższym kodzie występują dwie różne zmienne o tej samej nazwie `ile`: pierwsza wewnątrz definicji funkcji `zlicz` (zmienna lokalna), a druga na zewnątrz tej definicji (zmienna globalna). Na zewnątrz definicji funkcji nie możemy po prostu użyć zmiennej zadeklarowanej wewnątrz niej, ponieważ nie jest ona tam *widoczna*. Natomiast nazwanie dwóch zmiennych tak samo nie jest problemem. W takim przypadku na podstawie kontekstu Python zawsze wie, o którą zmienną chodzi.

- Uruchamiamy funkcję `zlicz` na wczytanym tekście i zapisujemy wynik jej działania w zmiennej `ile`
- Wypisujemy wartość zmiennej `ile`

Uruchom powyższy program, naciskając przycisk *run*. Jeśli się nie pomyliłeś, program powinien wypisać liczbę liter A występujących w pliku `tekst1.txt`.

Zadanie 1 Zmodyfikuj program tak, aby policzył, ile liter polskiego alfabetu (tj. AĄBCĆDEĘFGHIJKŁŁMNŃOÓPRSŚTUWYZŻŻ) występuje łącznie w pliku `tekst1.txt`.

Wskazówka: przy użyciu instrukcji `in` można sprawdzać, czy dana litera występuje wewnątrz danego słowa:

```
> 'N' in 'WYSZUKIWANIE'
True
> 'M' in 'WYSZUKIWANIE'
False
```

1.3 Zliczanie częstości występowania poszczególnych liter alfabetu

Aby zliczyć wystąpienia litery A, potrzebowaliśmy zmiennej `ile`, w której odnotowywaliśmy kolejne wystąpienia tej litery. Używając tej samej metody, moglibyśmy zliczyć wystąpienia każdej z 32 liter alfabetu z osobna, przy użyciu 32 zmiennych. Możemy to jednak zrobić wygodniej, korzystając z jednej 32-elementowej listy:

```
alfabet = 'AĄBCĆDEĘFGHIJKŁŁMNŃOÓPRSŚTUWYZŻŻ'

def zlicz(tekst):
    ile = [0] * 32
    for znak in tekst:
        if znak in alfabet:
            poz = alfabet.index(znak)
            ile[poz] = ile[poz] + 1
    return ile
```

W powyższym kodzie:

- Napis `[0] * 32` tworzy listę `[0, 0, ..., 0]` składającą się z 32 zer. Listę tę zapisujemy następnie w zmiennej `ile`.
- Dla każdego znaku z tekstu sprawdzamy, czy to litera. Jeśli tak, to obliczamy jej pozycję w alfabecie (przy użyciu instrukcji `index`) i zapisujemy otrzymaną pozycję w zmiennej `poz`.
- W liście `ile` zwiększamy wartość na pozycji `poz`, czyli wartość zliczającą liczbę wystąpień znaku `znak`.

Po wykonaniu powyższego kodu w zmiennej `ile` powinna znajdować się 32-elementowa lista zawierająca liczby wystąpień kolejnych liter alfabetu. Czyli na pierwszej pozycji powinna tam się znaleźć liczba liter A, na drugiej pozycji liczba liter A, itd. Aby sprawdzić, czy tak dzieje się naprawdę, zmodyfikuj program do następującej postaci:

```
alfabet = 'AĄBCĆDEĘFGHIJKŁŁMNŃOÓPRSŚTUWYZŻŻ'

def zlicz(tekst):
    ile = [0] * 32
    for znak in tekst:
        if znak in alfabet:
```

```

        poz = alfabet.index(znak)
        ile[poz] = ile[poz] + 1
    return ile

plik = open('tekst1.txt', 'r')
tekst1 = plik.read().upper()
plik.close()
ile = zlicz(tekst1)
print(ile)

```

Następnie uruchom ten program, klikając przycisk *run*. Program powinien wypisać 32-elementową listę liczb oznaczających liczbę wystąpień kolejnych liter alfabetu w pliku `tekst1.txt`.

2 Wizualizacja listy w formie wykresu

W poprzednim rozdziale udało nam się otrzymać listę liczby wystąpień poszczególnych liter alfabetu w tekście. Jak wiemy, takie zliczanie liter jest elementem *analizy częstościowej* — techniki łamania szyfrów tekstowych. Aby przeanalizować obliczone częstości, wygodnie jest przedstawić obliczoną listę w formie wykresu.

Do rysowania wykresów w Pythonie służy pakiet `pyplot`. Aby go używać, na początku programu musimy go zaimportować i wybrać odpowiedni tryb poleceniami:

```

import matplotlib as mpl
mpl.use('Agg')
import matplotlib.pyplot as plt

```

Teraz możemy już stworzyć wykres kolejnych wartości listy `ile` przy użyciu polecenia:

```
plt.plot(ile)
```

Aby oś była podpisana literami, możemy dodać:

```
plt.xticks(range(32), alfabet)
```

Na koniec, musimy zapisać wykres w pliku graficznym:

```
plt.savefig('tekst1.png')
```

Ostatecznie, nasz program do rysowania wykresu częstości występowania liter w tekście powinien wyglądać następująco:

```

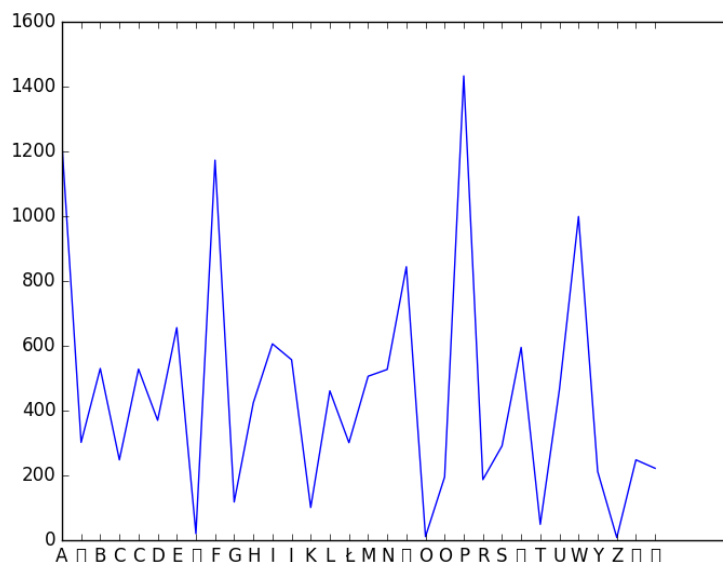
import matplotlib as mpl
mpl.use('Agg')
import matplotlib.pyplot as plt

alfabet = 'AĄBCĆDEĘFGHIJKLŁMNŃOÓPRSŚTUWYZŹŻ'

def zlicz(tekst):
    ile = [0] * 32
    for znak in tekst:
        if znak in alfabet:
            poz = alfabet.index(znak)
            ile[poz] = ile[poz] + 1
    return ile

plik = open('tekst1.txt', 'r')
tekst1 = plik.read().upper()

```



Rysunek 1: Wykres częstości występowania poszczególnych liter w tekście `tekst1.txt`

```
plik.close()
ile = zlicz(tekst1)

plt.plot(ile)
plt.xticks(range(32), alfabet)
plt.savefig('tekst1.png')
```

Po uruchomieniu powyższego kodu, po kilku sekundach w projekcie powinien pojawić się nowy plik o nazwie `tekst1.png` zawierający obrazek, taki jak na rys. 1.

2.1 Porównanie wykresu z częstościami występowania liter w języku polskim

Stwórz w swoim projekcie nowy plik o nazwie `lalka.txt` i skopiuj do niego zawartość pliku `lalka.txt` ze strony ciekawi.icm.edu.pl (zakładka *Dla uczestników* → *Materiały - ćwiczenia*). W pliku tym znajduje się fragment powieści *Lalka* Bolesława Prusa. Jest to fragment na tyle długi, że częstości występowania poszczególnych liter alfabetu w tym fragmencie dość dobrze odpowiadają średnim częstościom występowania poszczególnych liter w naszym języku.

Zadanie 2 Przy użyciu napisanego wcześniej programu naszkicuj wykres częstości występowania poszczególnych liter we fragmencie `lalka.txt` i zapisz go w pliku `lalka.png`.

Zadanie 3 Plik `tekst1.txt` powstał przez zaszyfrowanie pewnego tekstu napisanego w języku polskim. Porównaj wykres `tekst1.png` z wykresem `lalka.png`: wypisz 4 najczęściej i 4 najrzadziej występujące litery w obu tekstach. Czy na podstawie powyższej analizy częstościowej potrafisz wskazać szyfr, którego użyto?

3 Zadania dodatkowe

Zadanie 4 Narysuj oba wykresy (dla pliku `tekst1.txt` i dla pliku `lalka.txt`) na jednym rysunku. W tym celu należy dwa razy obliczyć odpowiednią listę częstości, a następnie dwa razy wykonać polecenie `plt.plot()`, oddzielnie dla każdej listy. Polecenia `plt.xticks()` i `plt.savefig()` wystarczy wykonać raz.

Zadanie 5 Jeśli poprawnie rozwiążesz poprzednie zadanie, okaże się, że jeden z wykresów jest dużo „niższy” niż drugi. Dzieje się tak, ponieważ jeden tekst jest dłuższy od drugiego, więc w jednym tekście jest więcej poszczególnych liter niż w drugim.

Zmodyfikuj funkcję **zlicz**, tak aby zliczała ona **względne** częstości występowania poszczególnych liter, tj. dla każdej litery ma zliczyć liczbę wystąpień tej litery podzieloną przez łączną liczbę wystąpień wszystkich liter.

Narysuj wykresy **względnych** częstości wystąpień liter w tekstach **tekst1.txt** i **lalka.txt** na jednym rysunku. Teraz oba wykresy powinny być podobnej wysokości.

Zadanie 6 Rozszyfruj tekst z pliku **tekst1.txt**. Narysuj wykres względnych częstości występowania poszczególnych liter w tym rozszyfrowanym tekście oraz w pliku **lalka.txt** na jednym rysunku. Wykresy powinny się mniej więcej pokrywać.

4 Praca domowa nr 3

Rozwiązania zadań należy przesłać do czwartku 20 kwietnia do godz. 16⁵⁹ na adres `licealisci.pracownia@icm.edu.pl` wpisując jako temat wiadomości `Lx PD3`, gdzie `x` to numer grupy, np. `L3 PD3` dla grupy L3, itd.

We wszystkich zadaniach będziemy korzystać z pliku **tekst2.txt** (dostępny do pobrania ze strony `ciekawicm.edu.pl`, zakładka *Dla uczestników* → *Materiały - ćwiczenia*).

Zadanie domowe 1 (1 pkt) Napisz program, który wczyta tekst z pliku **tekst2.txt**, zamieni wszystkie litery na wielkie i wypisze na ekran co piąty znak tego tekstu, począwszy od pierwszego (liczymy wszystkie znaki, również spacje i znaki interpunkcyjne).

Dla przykładu, początek pliku **tekst2.txt** wygląda tak:

Hgrłjńtzb eeół iee 1652. Jjprńwd pńc ggfoćp, pąż jłrcł; mćcp jayp fłmnbgbp Ngkuo...

Zatem Twój program powinien wypisać:

HŃEI6JWĆOPŁ B O...

Zadanie domowe 2 (2 pkt) Narysuj wykres częstości występowania poszczególnych liter w tekście będącym wynikiem zadania 1 (czyli w co piątym znaku pliku **tekst2.txt**). Wiedząc, że wykres ten odpowiada przesuniętemu wykresowi częstości występowania liter w naszym języku, znajdź to przesunięcie.

Wskazówka: jako wzorzec częstości występowania poszczególnych liter w naszym języku użyj częstości obliczonych dla pliku **lalka.txt**.

Zadanie domowe 3 (3 pkt) Plik **tekst2.txt** powstał w wyniku zaszyfrowania pewnego tekstu napisanego w języku polskim szyfrem Vigenere’a z pewnym pięcioliterowym kluczem. Znajdź ten klucz, a następnie rozszyfruj tekst.

Wskazówka: szyfr Vigenere’a z pięcioliterowym kluczem możemy traktować jak 5 szyfrów Cezara z 5 różnymi kluczami. W zadaniu 2 odkryliśmy pierwszą literę klucza — znalezione przesunięcie to indeks w alfabecie tej litery. Aby znaleźć drugą literę klucza, przeprowadź analogiczną procedurę dla co piątej litery tekstu począwszy od drugiej, itd.

Aby uzyskać 3 punkty za to zadanie, wielkość liter w rozszyfrowanym tekście powinna się pokrywać z oryginałem. Za rozwiązanie, które wypisuje rozszyfrowany tekst składający się z samych wielkich liter można uzyskać maksymalnie 2 punkty.