

LISTA ZADAŃ 2

Termin oddania: 21.04.2020

1. Porównywanie tekstów

Napisz program, który określi podobieństwo Jaccarda pomiędzy dwoma plikami tekstowymi. Program powinien reprezentować tekst jako zbiór n-gramów o określonej długości n. Podobieństwa Jaccarda dla zbiorów A, B definiuje się jako:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Program powinien jako argumenty wywołania przyjmować:

- nazwy plików tekstowych do analizy;
- przełącznik pomiędzy pracą na (a) literach lub (b) słowach jako podstawowych tokenach – czyli elementach składowych n-gramów, np. „Ala ma kota”:
(a) tokeny: ['a', 'k', 'l', 'm', 'o', 't'] 2-gramy: [„al”, „ko”, „la”, „ma”, „ot”, „ta”],
(b) tokeny: [„Ala”, „ma”, „kota”] 2-gramy: [„Ala ma”, „ma kota”];
- nazwę pliku z listą znaków przestankowych (np. spacja, interpunkcja) lub słów przestankowych (np. „i”, „ale”, „więc”), (łącznie nazywanych dalej tokenami przestankowymi);
- przełącznik włączający ignorowanie tokenów przestankowych; standardowo tokeny przestankowe nie wchodzi w skład n-gramów, ale przerywają ich rozbudowę, tzn. np. w napisie „Ala ma kota” nie ma 2-gramu „a<spacja>” lub „ak”);
- opcjonalną nazwę pliku do którego należy zapisać wszystkie wspólne n-gramy wraz z ich położeniem w plikach tekstowym (wiersz, pozycja w wierszu);
- opcjonalną długość n-gramu (jeśli nie określono, należy przyjąć 1).

Program powinien wypisać na standardowym wyjściu wartość podobieństwa Jaccarda.

Program należy przetestować znajdując rozsądne parametry użycia (liczba n) na

- sekwencjach białek (<https://www.ncbi.nlm.nih.gov/protein/>),
- abstraktach tekstach naukowych (<https://www.ncbi.nlm.nih.gov/pubmed/>).

2. Wielomian (programowanie obiektowe)

Zaprojektuj klasę Wielomian reprezentującą wielomian dowolnego stopnia. Klasa powinna zawierać metody implementujące typowe operacje na wielomianach (dodawanie, odejmowanie, mnożenie), zwracające stopień wielomianu oraz zwracające wartość wielomianu $W(x)$ dla zadanej wartości parametru x .

Kroki wprowadzające:

- sprawdź czy Twój język wspiera paradygmat obiektowy. Jeśli nie – oprogramuj rozwiązanie zgodnie z zasadami wybranego języki. Jeśli masz wątpliwości – skontaktuj się z prowadzącym;
- przypomnij sobie podstawowe pojęcia dotyczące wielomianów: współczynniki wielomianu, stopień wielomianu;
- przypomnij sobie podstawowe zasady wykonywania działań na wielomianach (dodawanie, odejmowanie, mnożenie wielomianów);
- przypomnij sobie, czym są pola i metody klasy;
- przypomnij sobie, czym jest konstruktor klasy;
- w języku Python zapoznaj się z działaniem metody `__init__` oraz parametru `self`;
- przypomnij sobie, czym są modyfikatory dostępu (np. `public`, `private`, `protected`);
- w języku Pythonie zapoznaj się z konwencją modyfikatorów dostępu;
- przypomnij sobie, czym są metody statyczne.

Zadania do wykonania:

- utwórz klasę Wielomian, której konstruktor przyjmuje listę współczynników wielomianu; zapewnij obsługę wyjątków (np. próby stworzenia wielomianu stopnia zerowego).
- zdefiniuj metodę zwracającą stopień wielomianu.
- zdefiniuj metodę zwracającą reprezentację tekstową wielomianu w postaci:
$$a_n * x^n + a_{n-1} * x^{n-1} + a_1 * x + a_0$$
- zdefiniuj metodę zwracającą wartość wielomianu $W(x)$ dla zadanej wartości parametru x .
- zdefiniuj metody wykonujące podstawowe operacje na wielomianach: dodawanie, odejmowanie, mnożenie wielomianów.
- (dodatkowe) przeładuj operatory dodawania, odejmowania i mnożenia (+, -, *) tak, aby reprezentowały odpowiednie operacje na wielomianach.
- W oddzielnym pliku napisz program, w którym zostaną utworzone obiekty klasy Wielomian i zaprezentowane działanie wszystkich zdefiniowanych metod oraz raportowanie wyjątków.

Witold Dyrka
Agnieszka Kazimierska