AAL - tablica mieszająca

Łukasz Pokorzyński (300251), Adam Steciuk (300263)
Listopad 2020

1 Wariant - W11 i W21

Przedmiotem analizy jest tablica mieszająca: tablica przechowuje rekordy zawierające napisy. Długość tablicy jest ograniczona arbitralnie przez pewną stałą K. Dla danego napisu s obliczamy k=M(s) gdzie M() jest funkcją mieszającą i umieszczamy strukturę reprezentującą napis w tablicy mieszającej: H[k]. W przypadku kolizji funkcji mieszającej (H[k] zajęte) reprezentujące napis s struktury danych zapisywane są w liście jednokierunkowej, której głowa to H[k]. Przedmiotem implementacji powinno być: dodanie i usunięcie elementów w H[]. Wybór funkcji mieszającej M(s) do decyzji projektanta.

Zastosować jedną funkcję mieszającą; dodatkowo przeprowadzić analizę dla enumeracji tablicy (wydobycia wszystkich elementów).

Testy przeprowadzić dla: sztucznie wygenerowanych słów, generator ma posługiwać się tablicą prawdopodobieństw wystąpienia danej litery na początku słowa (początek słowa) oraz litery po poprzedzającej literze, (spacja, kropka, przecinek, itp. traktowane są jako litera specjalna ”koniec słowa”). Prawdopodobieństwa należy uzyskać z próbki tekstu polskiego.

2 Podejście do problemu

Projekt zostanie zrealizowany w języku Python. Nie przewidujemy korzystania z bibliotek z funkcjami specyficznymi dla danego systemu operacyjnego, tak aby zapewnić pełną przenośność programu. Środowisko, którego będziemy używać do stworzenia projektu, obejmuje wersję 3.7 Pythona oraz program PyCharm. Do prezentacji wyników pomiarów skorzystamy z biblioteki matplotlib. Inne biblioteki, z których skorzystamy: random, re, collections.
Projekt składać się będzie z trzech zasadniczych części - generatora danych, zaimplementowanej tablicy mieszającej zgodnej z przypisanym wariantem oraz interfejsu użytkownika. Możliwe będzie zapisanie wygenerowanych danych do pliku oraz ich bezpośrednie przekazanie do programu. Zgodnie z wytycznymi, wykorzystywanymi strukturami danych będą tablica mieszająca oraz lista jednokierunkowa (ich dokładniejszy opis w dalszej części dokumentacji).
Trzema najistotniejszymi parametrami dla problemu analizowanego są:

- rozmiar tablicy mieszającej
- rozmiar danych wejściowych
- kolizyjność

Aby dokonać rzetelnej analizy oprócz przypadków standardowych weźmiemy przypadki skrajne takie jak mała tablica mieszająca względem dużej ilości danych, mała ilość danych i duża tablica, wprowadzanie dużej liczby tych samych słów itp. Z powodu użycia listy jednokierunkowej (o potencjalnie nieograniczonych rozmiarach) nie przewidujemy możliwości przepełnienia tablicy.

3 Generator danych testowych

W celu poprawnej obsługi polskich znaków zostanie przyjęte kodowanie UTF-8. Na początku próbka języka polskiego jest dzielona na słowa zgodnie z przedstawionymi wytycznymi. Generacja danych testowych opierać się będzie na dwóch słownikach: jeden będzie każdej literze przyporządkowywał liczbę słów nią się rozpoczynających a drugi będzie każdej literze przyporządkowywał słownik liter po

niej następujących wraz z liczbą ich wystąpień. W przypadku gdy litera kończy słowo, do słownika dodawany jest następnik w postaci pustego znaku (oraz analogicznie - liczba takich zdarzeń)

Generacja nowego słowa opiera się na wylosowaniu litery ze słownika liter rozpoczynających, używając średniej ważonej z wagami będącymi liczbą wystąpień danej litery na początku słów w próbce uczącej. Następnie dla wybranej litery losujemy literę ze słownika do niej przyporządkowanego, w ten sam sposób, używając średnich ważonych. Proces wybierania następników z kontynuujemy do momentu wylosowania pustego znaku, oznaczającego koniec słowa. Losowość generacji słów może być predeterminowana przez ziarno (seed).

Na podstawie wstępnych testów uznaliśmy, że próbka 1500 słów daje zadowalające wyniki.

Spodziewany sposób uruchomienia: generator in.txt l_słów out.txt [seed]
in.txt - plik wejściowy z próbką tekstu polskiego
out.txt - plik wyjściowy zwracający gotowe słowa
seed - opcjonalny parametr, możliwość przekazania ziarna do generacji słów
l_słów - liczba słów do wygenerowania

4 Lista jednokierunkowa

Każdy węzeł będzie obiektem posiadającym 2 pola: data - przechowującym właściwe informacje oraz wskazanie na następny węzeł w liście.
Klasa węzła posiadać będzie metody pozwalające na zwrócenie danych przechowywanych w węźle, zwrócenie wskazania na kolejny węzeł oraz ustawienie wskazania na podany w argumencie węzeł. Dane przechowywane w węźle nie będą modyfikowane w czasie istnienia obiektu, więc wystarczy możliwość przypisania ich poprzez konstruktor.

Lista jednokierunkowa będzie obiektem posiadającym wskazanie na pierwszy węzeł przechowujący dane. Będzie pozwalała na standardowe operacje takie jak dodawanie i usuwanie elementów, zwracanie wielkości (ile węzłów przechowuje w danym momencie), sprawdzanie czy dany element istnieje w liście, zwracanie swojej zawartości. Domyślnie lista będzie inicjalizowana z głową wskazującą na element ”None”. Ostatni węzeł nie wskazuje na kolejny, a na obiekt ”None”.

5 Tablica mieszająca

Tablica będzie korzystać z algorytmu mieszania opierającym się na sumowaniu wartości unicode liter słowa, a następnie wykonania dzielenia modulo przez długość tablicy w celu uzyskania pozycji, na której dane będą przechowywane. Tablica będzie obiektem składającym się z list jednokierunkowych, dla których głową będzie H[k], gdzie k to wartość hasha. Pozwoli na standardowe operacje na tablicy mieszającej, to znaczy dodawanie lub usuwanie elementu, obliczenie hasha dla danych wprowadzanych. Wielkość tablicy będzie możliwa do ustalenia przez użytkownika, nie będzie ona w żaden sposób ograniczona.

6 Interfejs użytkownika

Przewidujemy trzy możliwe sposoby uruchomienia programu:

- **Z dostarczeniem danych wejściowych** - użytkownik będzie miał możliwość przekazania do programu pliku tekstowego z sekwencją słów. Tryb ten umożliwi przetestowanie działania programu dla małych problemów.
  Spodziewany sposób uruchomienia: prog -m1 words.txt
- **Z generacją automatyczną** - użytkownik opcjonalnie będzie mógł ustawić seed do losowego

generowania słów, a następnie wykorzystania wytworzonych słów w programie.
Spodziewany sposób uruchomienia: prog -m2 in.txt l_słów [seed]

- **Z generacją i analizą wykonania** - standardowy tryb, który pozwoli na wygenerowanie słów oraz przeanalizowania działania algorytmów programu, wraz z wyświetleniem wyników (rozmiar problemu, czas trwania algorytmu). Pozwoli opcjonalnie na przetestowanie usuwania danych.
Spodziewany sposób uruchomienia: prog -m3 in.txt l_słów [seed] [delete.txt]
- **Z dostarczeniem danych wejściowych i analizą wykonania** - kolejny standardowy tryb, który pozwoli na przekazanie pliku z gotowymi danymi oraz przeanalizowania działania algorytmów programu, wraz z wyświetleniem wyników (rozmiar problemu, czas trwania algorytmu). Również opcjonalnie pozwoli na przetestowanie usuwania danych.
Spodziewany sposób uruchomienia: prog -m4 words.txt [delete.txt]

Parametry w sposobach uruchomienia:
-m* - tryby wykonania programu, w miejscu * cyfry od 1-4
words.txt - gotowe słowa uzyskane wcześniej od generatora
in.txt - plik wejściowy z próbką tekstu polskiego
seed - opcjonalny parametr, możliwość przekazania ziarna do generacji słów
delete.txt - opcjonalny parametr, pozwoli wykonać dodatkowe testy dla usuwania danych z tablicy
l_słów - liczba słów do wygenerowania

Włączenie programu ze specyficznym parametrem (na przykład -h) wyświetli informację o możliwych sposobach uruchomienia i ich krótkim opisem.

W przypadku niemożliwości otworzenia zadanego pliku, lub próby przetworzenia pliku zawierającego znaki w kodowaniu niekompatybilnym z UTF-8, program będzie kończył swoje działanie zwracając kod informujący o zaistniałym błędzie.