

Sprawozdanie z Laboratorium 3 - Pomiar czasu znajdywania losowego słowa z listy. Implementacja listy, stosu oraz kolejki przy wykorzystaniu odpowiednich interfejsów.

Kamil Kuczaj

30 marca 2016

1 Wstęp

Podanym zadaniem był pomiar czasu znajdywania losowego elementu listy typu *string*. Należało wykonać pomiary zapisu: 10^1 , 10^3 oraz 10^5 . Wykorzystano słownik 109582 najpopularniejszych słów w języku angielskim. Zdecydowano się na wybór języka angielskiego nad polskim z uwagi na zlikwidowanie problemów z wczytywaniem znaków łacińskich. W programie zaimplementowano szablony, dzięki czemu bardzo łatwo jest zmierzyć czas dla innych zmiennych lub nawet własnych klas.

Słowa w słowniku posortowane są alfabetycznie, więc za każdym razem do struktury danych były wczytywane będąc już uprzednio posortowane rosnąco.

2 Specyfikacja komputera

Wersja kompilatora <i>g++</i>	4.8.4
System	Ubuntu 14.04.4
Procesor	Intel Core i5 2510M 2.3 GHz
Pamięć RAM	8 GB DDR3 1600 MHz
Rozmiar zmiennej <i>int</i>	4 bajty

3 Pomiary oraz ich interpretacja

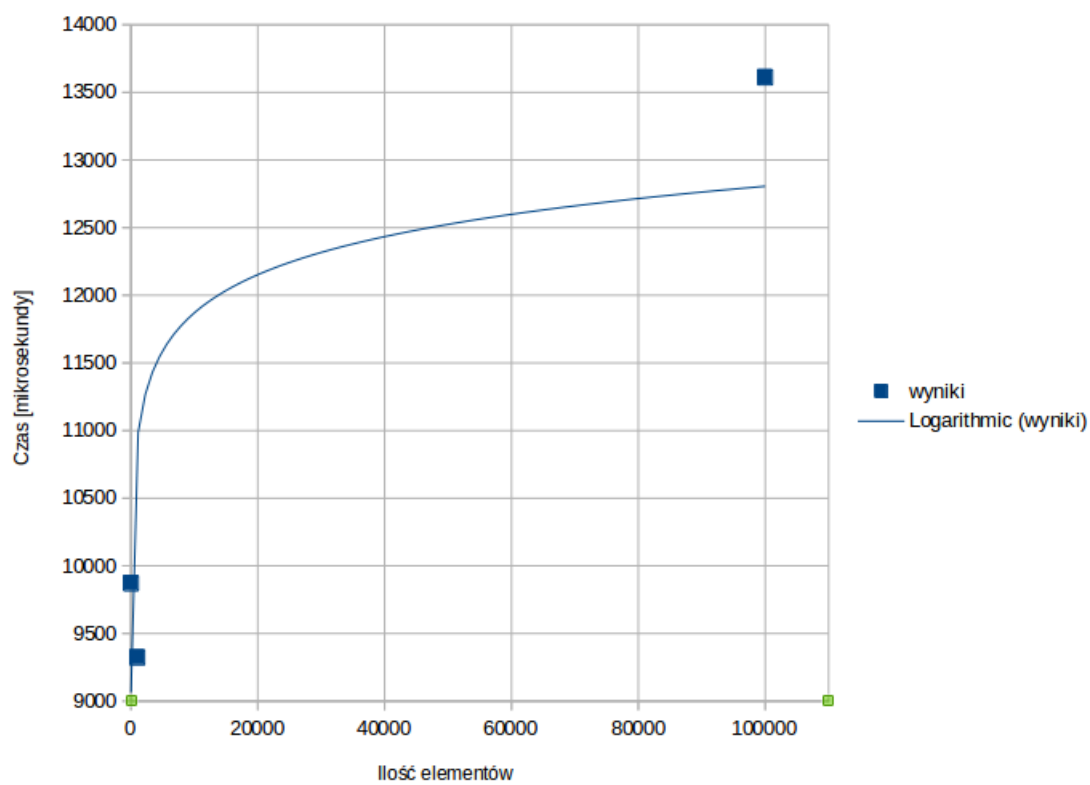
Bardzo dużą część czasu zajmowało zapisanie stu tysięcy słów - wynikało to z tego, że należało zaalokować pamięć, w tym wypadku zwiększając ją o jeden element za każdym razem. Wiąże się to ze złożonością obliczeniową rzędu n^2 w notacji dużego Θ .

W programie użyłem funkcji *rand()* z biblioteki *<cstdlib>* do wylosowania losowego słowa ze słownika 109582 najpopularniejszych słów w języku angielskim. Następnie przeszukiwałem listę w poszukiwaniu tego słowa i zapisywałem go do pliku niezależnie od tego, czy słowo zostało znalezione. Chciałem w ten sposób pokazać, że przeszukanie listy jest bardzo efektywne. Poniżej w tabeli przedstawione zostały średnie wyniki pomiarów.

Ilość elementów	10^1	10^3	10^5
Średni czas [μ s]	9871,2	9323,78	13611,46
Średni czas [s]	0,0098712	0,00932378	0,01361146

Tablica 1: Wyniki pomiarów. Łatwo zauważyć, że przeszukanie listy trwa bardzo krótko i prawie nie zależy od ilości elementów

Wydawać by się mogło, że liczenie średniej z pomiarów, gdzie nie uwzględnia się czy słowo zostało znalezione, lub gdzie się znajdowało może być niedokładne. Jednak wyraźnie widać, że to skomplikowanie pomiarów miało by nieznaczny wpływ na uzyskane wyniki.



Rysunek 1: Zobrazowanie wyników pomiaru oraz regresja logarytmiczna wykonana w programie *LibreOffice Calc*

4 Wnioski

Zastosowanie struktury danych typu *lista* zdecydowanie pozwoliło zmniejszyć czas na znalezienie słowa. Dysponując podanymi danymi zakwalifikowałbym przeszukanie tablicy jako algorytm rzędu $\log(n)$ w notacji dużego Θ .