

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

Kamil Kuczaj

20 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.

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

```
g++ -Iinc -Wall -pedantic -g -c -o obj/main.o src/main.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Stoper.o src/Stoper.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Sedzia.o src/Sedzia.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Tablica.o src/Tablica.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Stos.o src/Stos.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Kolejka.o src/Kolejka.cpp
g++ obj/main.o obj/Stoper.o obj/Sedzia.o obj/Tablica.o obj/Stos.o obj/Kolejka.o obj/Lista.o -o program
./program
!!! INFO: Program starts indexing at 0, not at 1 !!!

Found: aahed at index: 2 in 10-sized array in time of: 0.003 milliseconds
Found: abate at index: 42 in 1000-sized array in time of: 0.009 milliseconds
Found: mishap at index: 59984 in 100000-sized array in time of: 1.989 milliseconds
```

Rysunek 1: Pomiar czasu znajdowania losowego słowa.

```
g++ -Iinc -Wall -pedantic -g -c -o obj/main.o src/main.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Stoper.o src/Stoper.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Sedzia.o src/Sedzia.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Tablica.o src/Tablica.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Stos.o src/Stos.cpp
g++ -Iinc -Wall -pedantic -g -c -o obj/Kolejka.o src/Kolejka.cpp
g++ obj/main.o obj/Stoper.o obj/Sedzia.o obj/Tablica.o obj/Stos.o obj/Kolejka.o obj/Lista.o -o program
./program
!!! INFO: Program starts indexing at 0, not at 1 !!!

Found: aardvark at index: 5 in 10-sized array in time of: 0.003 milliseconds
Found: accreditation at index: 685 in 1000-sized array in time of: 0.06 milliseconds
Found: sightlier at index: 87783 in 100000-sized array in time of: 2.645 milliseconds
```

Rysunek 2: Pomiar czasu znajdowania innego losowego słowa.

Bardzo dużą część czasu zajmowało zapisanie stu tysięcy słów - wynikało to z tego, że należało je zaalokować, a potem wylosować z tej puli. Przeszukiwanie listy było natomiast bardzo krótkie i zwykle nie przekraczało 3 milisekund.

4 Wnioski

Zastosowanie struktury danych typu lista zdecydowanie pozwoliło zmniejszyć czas na znalezienie słowa. Zdecydowanie pomogło mi również w tym jak ważne jest zastosowanie odpowiedniej struktury danych do programu.