

Programowanie II R

zestaw zadań 5

Obiekty z biblioteki STL part 2: set, map

Strumienie, zapis do pliku

10 kwietnia 2025

Strona wykładowcy <http://glach.wikidot.com/p2r>
(prawdopodobnie będzie aktualizowana w przyszłości)

—Źródła do STL: pojemniki oraz iteratory—

Dla informacji strony: STL pojemniki i kiedy ich używać:

Pojemniki z STL

Jak używać iteratorów: Iteratory

Zadanie 1: Szukanie unikalnych słów za pomocą pojemnika set

Mając pastę o rick and morty z pliku "pastarickandmorty.txt" użyj pojemnika set z

```
#include<set>
```

żeby wychwycić słownictwo używane w tej paście. Do otwarcia pliku użyj ifstream. Do przejścia się pętlą po zbiorze użyj iteratora. Spróbuj dodać słowo, które już istnieje w zbiorze i zobacz co się stanie.

Podpowiedź: wykorzystaj `ispunct()` oraz `tolower()`. Możesz wykorzystać dla skrócenia kodu `stringstream`. Można też zastąpić funkcję używając biblioteki `algorithm` (jak chcesz możesz poszukać jak).

Zadanie 2: Szukanie słowa w słowniku

Tu dalej porównujemy jak szybko działają różne pojemniki na dane, tutaj kontekście szukania w mapie a szukania w wektorze.

```
https://raw.githubusercontent.com/dwyl/english-words/master/words.txt
```

1. Pobierz powyższy plik i za pomocą pojemnika "map" na struct zawierający int i string (na potrzebę zadania możesz zrobić trochę bardziej złożoną strukturę/mapę) utwórz słownik. Jako kluczy użyj słów (w sensie stringów, które są słowami). Do otwarcia pliku użyj ifstream.
2. Wypisz 4 najdłuższe słowa zaczynające się na "z" oraz 3 najkrótsze słowa zaczynające się na "a". Zmierz czas potrzebny do tego.
3. Zmierz czas potrzebny do znalezienia i wypisania słowa "Mjollnir".
4. Zrób punkty 2 i 3 tylko za pomocą `vector<string>` gdzie indeks w wektorze służy za numer linii. Do przejścia się pętlą po wektorze użyj iteratora.
O iteratorach raz jeszcze
5. W sposób losowy zmień kolejność słów w wektorze, żeby było nie po kolei. Następnie posortuj wektor używając "sort" z `#include<algorithm>`

Zadanie 3: Palindromy (operacje na string i zapis do wielu plików)

Korzystając z poprzedniego pliku z zadania 5 z zestawu 4 utwórz program znajdujący i wypisujący palindromy.

- Najpierw zrób wersję, gdzie wyszukuje palindromy ściśle to znaczy odbicia lustrzane litera do litera.
- Utwórz podobną funkcję która wyszukuje palindromy, które nie mają wszystkich liter tej samej wielkości z prawej i z lewej strony odbicia.
- Wypisz palindromy jednego i drugiego typu do osobnych plików używając biblioteki `fstream`.

Podpowiedź: możesz wykorzystać fakt, że string jest tablicą na `char`.

(Jakby mało tu było zadań to możesz wrócić do poprzednich serii, zadań z gwiazdką i/lub spytać o więcej)