

Programowanie II R

zestaw zadań 4

Obiekty z biblioteki STL, iteratory, powtórzenie z pętli

3 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(wstępne, pokazowe) Powtórzenie wektorów wraz z iteratorami oraz pętlami

Utwórz wektor z `vector` na `double` przy czym wypełnij go używając

1. iteratora dereferencyjnego z użyciem `auto` w pętli `for`, w tym również spróbować wypisać
2. iteratora odwrotnego (analogicznie)
3. iteratora stałego
4. Wypisać wektor do konsoli bądź pliku za pomocą składni "pythonowej"

Teraz spróbuj wykorzystać do tego samego pętle:

1. `while`
2. `do ... while`
3. pętli `for`, ale nie posiadającej warunku ukończenia, w jaki sposób zakończyć taką pętlę?

Zadanie 2: Dostęp losowy vs ustrukturyzowany

Wykorzystując bibliotekę `chrono` porównaj szybkość umieszczenia obiektu w środku (losowy dostęp) do umieszczenia elementu na końcu obiektu:

1. obiektu `vector` (`push back`)
2. obiektu `deque` (tu można od `push front` i `push back`)

Zadanie 3:Wykorzystanie bibliotek `numeric`, `functional` i `algorithm`

1. Utwórz wektor o długości 100000 i wypełnij go używając `iota` z `<numeric>`.
2. Potasuj wektor używając `shuffle` z `<algorithm>`.
3. Posortuj wektor używając `sort` z STL'a najpierw od najmniejszej do największej i odwrotnie, zmierz mu czas sortowania. Użyj `greater` z

```
#include<functional>
```

4. wypisz wektor używając iteratora
5. *[dla chętnych] napisz `quicksort`(iteracyjny bądź rekurencyjny) i porównaj czas sortowania z sortem z STL'a.

Zadanie 4: 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 przejścia się pętlą po zbiorze użyj iteratora. Spróbuj dodać słowo, które już istnieje w zbiorze i zobacz co się stanie.

Zadanie 5: Szukanie słowa w słowniku

<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).
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>`
Pomóż sobie kodem z poprzedniego zadania do punktów 3 i 4.