Lab02 - podstawowe struktury danych

Mateusz Krawczuk, nr indeksu 209147

Wygenerowano przez Doxygen 1.8.6

Cz, 19 mar 2015 01:08:41

Część I

Streszczenie

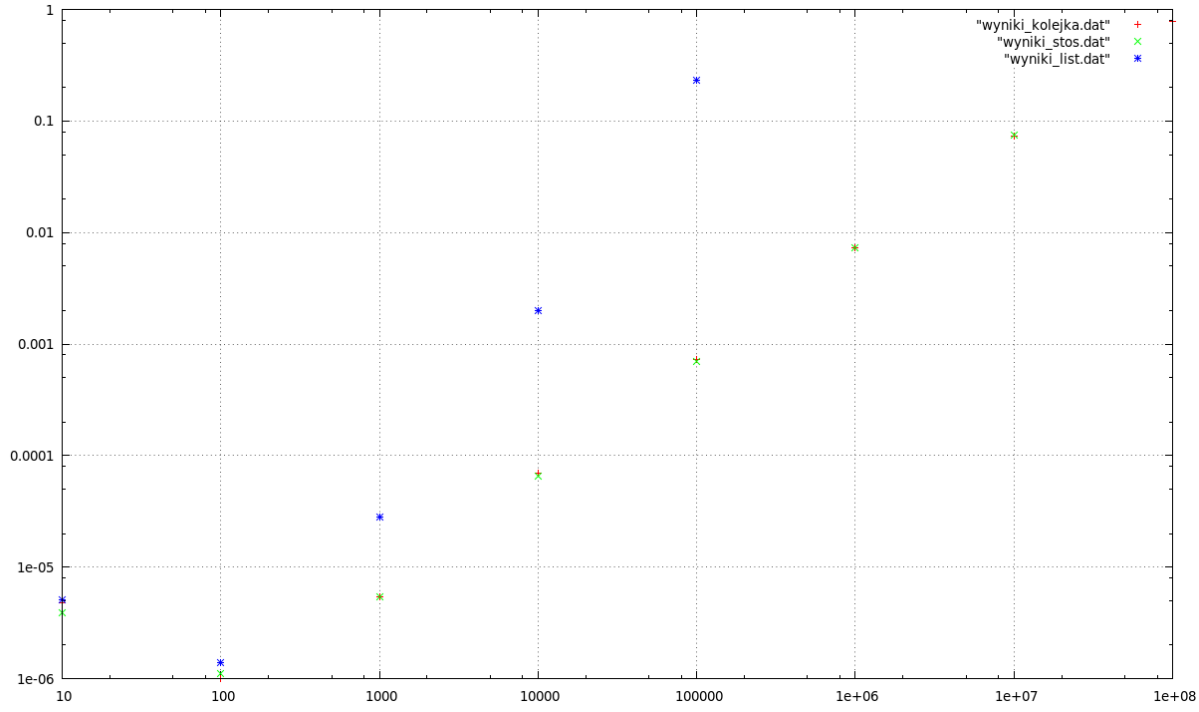
Niniejszy dokument zawiera wyniki pomiaru czasu, którego potrzebował mój komputer na zapełnienie zaimplementowanych przeze mnie struktur danych: stos, kolejka oraz lista zestawami danych o długościach od 1 do 10e7 elementów. Zawiera także dokumentację kodu, który pojawił się w projekcie od poprzedniego sprawozdania.

Część II

Sprawozdanie

Obliczenia wykonano na 64-bitowym procesorze AMD Athlon X2. Wykres przedstawia zależnośćczasu wykonywania operacji od długości ciągu danych. Podziałki na obydwu osiach są w skali logarytmicznej. Zależności czasowe dla zapełniania stosu i kolejki są prawie identyczne i ich złożoność jest O(n); wynika to z obranego sposobu implementacji - klasy reprezentujące te struktury danych są pochodnymi pewnej klasy bazowej, po której dziedziczą metodę ładowania do struktury, a więc dla obydwu baz danych proces przebiega identycznie. Mimo to przeprowadzono badanie osobno dla każdej z tych baz danych. Gdyby badnie dotyczyło zdejmowania elementów z tych struktur danych, wyniki z pewnością różniłyby się.

Klasa Kontener, która miała stanowić bazę do implementacji zadanych struktur przybrała w trakcie tworzenia formę listy jednokierunkowej, do której ładowanie danych okazało się ciężkim zadaniem dla komputera. Lista powinna zezwalać na dodawanie i usuwanie elementów w dowolnym miejscu i pod takim właśnie kątem została zbadana. Symulacje pokazały, że czas wykonywania operacji już dla niewielkich zestawów danych rośnie w bardzo dużym tempie. To pozostawia szerokie pole do optymalizacji programu



Część III

Dokumentacja klas

1 Dokumentacja struktury cegla

Struktura pomocnicza cegla.

`#include <cegla.h>`

Atrybuty publiczne

- int dana
- cegla * nastepna

1.1 Opis szczegółowy

Struktura reprezentuje elementarną cząstkę kontenera - zawiera daną oraz wskaźnik do kolejnego elementu w kontenerze.

Definicja w linii 15 pliku cegla.h.

1.2 Dokumentacja atrybutów składowych

1.2.1 int cegla::dana

Definicja w linii 17 pliku cegla.h.

1.2.2 cegla* cegla::nastepna

Definicja w linii 18 pliku cegla.h.

Dokumentacja dla tej struktury została wygenerowana z pliku:

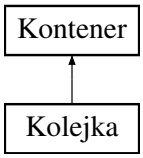
- cegla.h

2 Dokumentacja klasy Kolejka

Kolejka, abstrakcyjna struktura danych z buforem typu LIFO.

`#include <kolejka.h>`

Diagram dziedziczenia dla Kolejka



Metody publiczne

- int pop ()

  Pop kolejki jaki jest, każdy widzi. Usuwa najstarszy element i zwraca wartość przez niego przechowywaną.

Dodatkowe Dziedziczone Składowe

2.1 Opis szczegółowy

Definicja w linii 12 pliku kolejka.h.

2.2 Dokumentacja funkcji składowych

2.2.1 int Kolejka::pop ( ) [inline]

Definicja w linii 18 pliku kolejka.h.

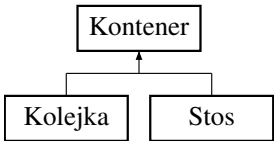
Dokumentacja dla tej klasy została wygenerowana z pliku:

- kolejka.h

3 Dokumentacja klasy Kontener

`#include <kontener.h>`

Diagram dziedziczenia dla Kontener



Metody publiczne

- Kontener ()
  Konstruktor klasy Kontener.
- void push (int)
  Wrzuca nową cegiełkę na początek kontenera.
- void insert (int wartosc, int indeks)
  Umieszcza cegłę z wartością 'wartosc' w miejscu oddalonym o 'indeks' miejsc od początku kontenera.
- int size ()
  Liczy z ilu cegiełek składa się kontener.
- int erase (int)
  Usuwa z listy element o wybranym indeksie.
- int find (int)
  Odnajduje w kontenerze przekazaną w argumencie wartość.
- void show ()
  Wypisuje elementy listy od najmłodszego zaczynając.

Atrybuty chronione

- cegla * alfa

3.1 Opis szczegółowy

Definicja w linii 13 pliku kontener.h.

3.2 Dokumentacja konstruktora i destruktora

3.2.1 Kontener::Kontener ( ) [inline]

Konstruktor klasy Kontener inicjalizuje wskaźnik 'alfa' wartością NULL.

Definicja w linii 25 pliku kontener.h.

3.3 Dokumentacja funkcji składowych

3.3.1 int Kontener::erase (int *indeks*)

Zainicjalizowane są dwa wskaźniki na najmłodszą cegiełkę. Jeden jest ustawiany na element do usunięcia, drugi na element o jeden młodszy. Następuje roszada wskaźników: wskaźnik młodszej cegły wskazuje na cegłę starszą od usuwanej, zostaje zapisana wartość przechowywana przez usuwaną cegłę i wreszcie zwolniona zostaje pamięć zajmowana dotychczas przez cegłę.

Zwraca

Zwraca wartość przechowywaną w usuniętej cegiełce.

Definicja w linii 58 pliku kontener.cpp.

3.3.2 int Kontener::find (int *wartosc*)

Powołany do życia jest szpieg - wskaźnik na obiekt typu 'cegla', który przemierza kontener w poszukiwaniu najwcześniejszego wystąpienia poszukiwanej wartości. Operacji towarzyszy licznik, który śledzi ilość miniętych przez wskaźnik cegieł, którą metoda zwraca. Należy ją interpretować jako indeks cegły, gdzie najwcześniej wystąpiła poszukiwana wartość.

Zwraca

Zwraca indeks (liczony od 0 od najmłodszej cegiełki) najbliższego wystąpienia poszukiwanej wartości.

Definicja w linii 91 pliku kontener.cpp.

3.3.3 void Kontener::insert (int *wartosc,* int *indeks*)

UWAGA: Indeks liczony jest od zera, od najmłodszej cegły.

Definicja w linii 15 pliku kontener.cpp.

3.3.4 void Kontener::push (int *wart*)

Definicja w linii 7 pliku kontener.cpp.

3.3.5 void Kontener::show ( )

Definicja w linii 108 pliku kontener.cpp.

3.3.6 int Kontener::size ( )

Zainicjalizowany zostaje wskaźnik na strukturę 'cegla' wskazujący na najmłodszą cegiełkę. Zostaje on potem wysłany w epicką podróż na sam koniec kontenera (czyli do napotkania NULLa). Towarzyszy mu licznik, który zlicza mijane po drodze cegiełki, których ilosc funkcja zwraca.

Zwraca

Zwraca wielkość kontenera.

Definicja w linii 43 pliku kontener.cpp.

3.4 Dokumentacja atrybutów składowych

3.4.1 cegla* Kontener::alfa [protected]

Definicja w linii 16 pliku kontener.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

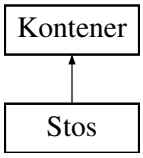
- kontener.h
- kontener.cpp

4 Dokumentacja klasy Stos

Stos, abstrakcyjna struktura danych z buforem typu FIFO.

`#include <stos.h>`

Diagram dziedziczenia dla Stos



Metody publiczne

- int pop ()

  Pop stosu jaki jest, każdy widzi. Zdejmuje najmłdoszy element i zwraca wartość przez niego przechowywaną.

Dodatkowe Dziedziczone Składowe

4.1 Opis szczegółowy

Definicja w linii 12 pliku stos.h.

4.2 Dokumentacja funkcji składowych

4.2.1 int Stos::pop ( ) [inline]

Definicja w linii 18 pliku stos.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- stos.h

Część IV

Dokumentacja plików

5 Dokumentacja pliku cegla.h

Plik zawiera definicję struktury pomocniczej cegla.

Komponenty

- struct cegla

  Struktura pomocnicza cegla.

6 Dokumentacja pliku kolejka.h

Plik zawiera definicję klasy Kolejka jako pochodnej klasy Kontener oraz definicje jego metod.

```
#include "kontener.h"
```

Komponenty

- class Kolejka

  Kolejka, abstrakcyjna struktura danych z buforem typu LIFO.

7 Dokumentacja pliku kontener.h

Plik zawiera definicję klasy Kontener oraz deklaracje metod tej klasy.

```
#include <cstdlib>
#include <iostream>
#include "cegla.h"
```

Komponenty

- class Kontener

8 Dokumentacja pliku stos.h

Plik zawiera definicję klasy Stos jako pochodnej klasy Kontener oraz definicje jego metod.

```
#include "kontener.h"
```

Komponenty

- class Stos

  Stos, abstrakcyjna struktura danych z buforem typu FIFO.