program framework benchmarkujacy dla abstrakcyjnego typu danych: tablica haszujaca.

1.0

Wygenerowano przez Doxygen 1.8.9.1

Śr, 15 kwi 2015 14:45:29

Spis treści

1 program framework benchmarkujacy dla abstrakcyjnego typu danych: tablica haszujaca.

Autor

Wojciech Makuch

Data

15.04.2015

Wersja

1.0

Program realizuje zadanie wyliczenia zlozonosci obliczeniowej dla abstakcyjnego typu danych jakim jest tablica haszujaca Uzyskane wyniki program zapisuje do pliku o nazwie *pomiar_czasu_5.txt*. *Program* zbudownay na klasach: dane - przechowujace klucz oraz wartosci danych, tablica - tworzacy tablice haszujaca(asocjacyjna z mozliwoscia przeszukania) Zlozonosc obliczeniowa liczona na podstawie wartosci lodowych otrzymanych z funkcji rand().

2 Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

3 Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

4 Dokumentacja klas

4.1 Dokumentacja klasy dane

`#include <dane.hh>`

Metody publiczne

- char * WezKlucz () const

  definicja metody WezKlucz klasy dane metoda nie pozwala na zmiane zawartosci pola.
- int WezWartosc () const

  definicja metody WezWartosc klasy dane. metoda nie pozwala na zmiane zawartosci pola.
- dane (char *k, int w)

  definicja konstruktora 2-parametrycznego
- dane ()

  definicja konstruktora bezparametrycznego zeruje pole wartosc, wskaznik na lancuch ustawia na NULL

4.1.1 Opis szczegółowy

Definicja w linii 16 pliku dane.hh.

4.1.2 Dokumentacja konstruktora i destruktora

4.1.2.1 dane::dane (char * *k*, int *w*) [inline]

Definicja w linii 38 pliku dane.hh.

4.1.2.2 dane::dane ( ) [inline]

Definicja w linii 44 pliku dane.hh.

4.1.3 Dokumentacja funkcji składowych

4.1.3.1 char* dane::WezKlucz ( ) const [inline]

Zwraca

pole prywane klucz.

Definicja w linii 26 pliku dane.hh.

4.1.3.2 int dane::WezWartosc ( ) const [inline]

Zwraca

pole prywatne wartosc

Definicja w linii 33 pliku dane.hh.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- dane.hh

4.2 Dokumentacja klasy tablica

definicja klasy tablica. Definiuje tablice haszujaca(asocjacyjna z mozliwoscia odwolania sie do jej elementu poprzez klucz). Tablica alokowana jest dynamicznie, rozmiar ustalany jest w konstruktorze. Tablica ma narzucony maksymalny rozmiar, poniewaz metoda haszujaca klucze na nim bazuje. Klasa zawiera 2 pola: tab - tablice danych oraz zmienna rozmiar przechowujaca informacje o dlugosci tablicy.

`#include <haszowanie.hh>`

Metody publiczne

- tablica (int n)

  definicja konstruktra 1-parametrowego
- ∼tablica ()

  definicja destruktora
- dane operator[ ] (int i) const

  defiicja przeciazenia operatora []
- dane operator[ ] (char ∗klucz) const

  definicja przeciazenia operatora [] pozwala na odwolanie sie do elementu tablicy za pomoca klucza
- int Haszowanie (char ∗klucz) const

  definicja metody Haszowanie pierszy sposob haszowania klucza dodaje elementu klucza zrzutowane na inta, liczy reszte z dzielenia przez maksymalny rozmiar metoda nie zmienia stanu obiektu
- int DrugieHaszowanie (char ∗klucz) const

  definicja metody DrugieHaszowanie drugi sposob haszowania klucza. wykorzystuje wzor 1-(dodany wartosci klucza)/q, gdzie q jest polowa maksymalnego rozmiaru i q jest liczba nieparzysta. metoda nie zmienia stanu obiektu. DrugieHaszowanie wykorzystuje sie w celu zminimalizowania ilosci wystapien kolizji.
- void Wypelnij (dane element)

  definicja metody Wypelnij Ustawia element w tablicy haszowania zgodnie z wartoscia klucza Jezeli dojdzie do 1. kolizji, zostaje wykorzystane drugie haszowanie i sprawdzana jest dostepnosc komorki i+j, gdzie i-wynik pierwszego haszowania, j - wynik drugiego haszowania. Jezeli dochodzi do kolejnych kolizji tablica zostaje wypelniana poprzez przechodzenie z komorki do komorki o 1 dalej.
- void Wyswietl () const

  definicja metody Wyswietl metoda pomocna przy tworzeniu programu. Wypisuje na strumien wyjsciowy zawartosc tablicy, przy odwolaniu sie do wartosci elementow(a nie do kluczy). Metoda nie zmienia stanu obiektu.

4.2.1 Opis szczegółowy

Definicja w linii 19 pliku haszowanie.hh.

4.2.2 Dokumentacja konstruktora i destruktora

4.2.2.1 tablica::tablica (int *n*)

Definicja w linii 4 pliku haszowanie.cpp.

4.2.2.2 tablica::∼tablica ( )

Definicja w linii 10 pliku haszowanie.cpp.

4.2.3 Dokumentacja funkcji składowych

4.2.3.1 int tablica::DrugieHaszowanie (char ∗ *klucz*) const

Parametry

[klucz]	zadany klucz

Zwraca

kod bedacy wynikiem drugiego haszowania

Definicja w linii 34 pliku haszowanie.cpp.

4.2.3.2 int tablica::Haszowanie (char ∗ *klucz*) const

Parametry

[klucz]	zadany klucz

Zwraca

kod bedacy wnikiem pierwszego haszowania

Definicja w linii 15 pliku haszowanie.cpp.

4.2.3.3 dane tablica::operator[ ] (int *i*) const [inline]

Parametry

[i]	komorka tablicy pozwala na odwolanie sie do elementu tablicy o indeksie i metoda nie pozwala na zmiane wartosci pol prywatnych

Zwraca

element tablicy

Definicja w linii 41 pliku haszowanie.hh.

4.2.3.4 dane tablica::operator[ ] (char ∗ *klucz*) const

Parametry

[klucz]	zadany klucz

Zwraca

element tablicy

Definicja w linii 96 pliku haszowanie.cpp.

4.2.3.5 void tablica::Wypelnij (dane *element*)

w

Parametry

[element]	dane przekazywane do tablicy

Definicja w linii 57 pliku haszowanie.cpp.

4.2.3.6 void tablica::Wyswietl ( ) const

Definicja w linii 87 pliku haszowanie.cpp.

Dokumentacja dla tej klasy została wygenerowana z plików:

- haszowanie.hh
- haszowanie.cpp

5 Dokumentacja plików

5.1 Dokumentacja pliku benchmark.cpp

```
#include <windows.h>
#include <ctime>
#include <fstream>
#include <iostream>
#include "losowy_lancuch.hh"
#include "dane.hh"
#include "haszowanie.hh"
```

Funkcje

- LARGE_INTEGER startTimer ()

  definicja funkcji StartTimer Rozpoczyna pomiar czasu Funkcja pobrana ze strony http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/
- LARGE_INTEGER endTimer ()

  definicja funkcji endTimer Konczy pomiar czasu Funkcja pobrana ze strony http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/
- double zliczaj_czas (int N)

  definicja funkcji zliczaj_czas funkcja benchmarkujaca zliczajaca czas wypelnienia tablicy haszujacej elementami losowymi. funkcja tworzy obiekt zadajac mu maksymalny rozmiar wynoszacy 150 000.
- int benchmarkuj (char ∗nazwa_pliku)

  definicja funkcji benchmarkuj Tworzy plik o zadanej nazwie. Wykonuje w petlach funkcje zliczaj czas, wyniki wysyla odpowiednio do strumienia wyjsciowego oraz do pliku. Po zakonczeniu dzialania funkcja zamyka plik

5.1.1 Dokumentacja funkcji

5.1.1.1 int benchmarkuj (char ∗ *nazwa_pliku*)

Parametry

[nazwa_pliku]	nazwa pliku otwartego do zapisu

Zwraca

0, gdzy funkcja zostanie wyonana poprawie
-1, w przypadku przeciwnym.

Definicja w linii 50 pliku benchmark.cpp.

5.1.1.2 LARGE_INTEGER endTimer ( )

Definicja w linii 19 pliku benchmark.cpp.

5.1.1.3 LARGE_INTEGER startTimer ( )

Definicja w linii 10 pliku benchmark.cpp.

5.1.1.4 double zliczaj_czas (int *N*)

Parametry

[N]	ilosc elementow wypelniajacych tablice

Zwraca

czas wypelniania elementami losowymi

Definicja w linii 30 pliku benchmark.cpp.

5.2 Dokumentacja pliku benchmark.hh

definicje funkcji zliczajacych czas operacji wypelnienia tablic haszujacych

```
#include <windows.h>
#include <ctime>
#include <fstream>
#include "losowy_lancuch.hh"
#include "dane.hh"
#include "haszowanie.hh"
```

Funkcje

- LARGE_INTEGER startTimer ()

  definicja funkcji StartTimer Rozpoczyna pomiar czasu Funkcja pobrana ze strony http://jaroslaw.↩mierzwa.staff.iiar.pwr.wroc.pl/
- LARGE_INTEGER endTimer ()

  definicja funkcji endTimer Konczy pomiar czasu Funkcja pobrana ze strony http://jaroslaw.mierzwa.↩staff.iiar.pwr.wroc.pl/
- double zliczaj_czas (int N)

  definicja funkcji zliczaj_czas funkcja benchmarkujaca zliczajaca czas wypelnienia tablicy haszujacej elementami losowymi. funkcja tworzy obiekt zadajac mu maksymalny rozmiar wynoszacy 150 000.
- int benchmarkuj (char ∗nazwa_pliku)

  definicja funkcji benchmarkuj Tworzy plik o zadanej nazwie. Wykonuje w petlach funkcje zliczaj czas, wyniki wysyla odpowiednio do strumienia wyjsciowego oraz do pliku. Po zakonczeniu dzialania funkcja zamyka plik

5.2.1 Dokumentacja funkcji

5.2.1.1 int benchmarkuj (char ∗ *nazwa_pliku*)

Parametry

[nazwa_pliku]	nazwa pliku otwartego do zapisu

Zwraca

0, gdzy funkcja zostanie wyonana poprawie
-1, w przypadku przeciwnym.

Definicja w linii 50 pliku benchmark.cpp.

5.2.1.2 LARGE_INTEGER endTimer ( )

Definicja w linii 19 pliku benchmark.cpp.

5.2.1.3 LARGE_INTEGER startTimer ( )

Definicja w linii 10 pliku benchmark.cpp.

5.2.1.4 double zliczaj_czas (int *N*)

Parametry

[N]	ilosc elementow wypelniajacych tablice

Zwraca

czas wypelniania elementami losowymi

Definicja w linii 30 pliku benchmark.cpp.

5.3 Dokumentacja pliku dane.hh

plik przechowujacy deklaracje klasy dane oraz deklaracje jej pol i metod.

Komponenty

- class dane

5.3.1 Opis szczegółowy

/∗! definicja klasy dane. Jest to struktura przechowujaca klucz poddawany haszowaniu oraz wartosc jako przechowywane dane (zmienna typu int). Obiekty typu dane sa wykorzystywane do wypelniania tablicy haszujacej(asocjacyjnej). Pole klucz mozna potraktowac np. jako nazwisko szukanej osoby w bazie danych, a pole wartosc jako jej nr telefonu. Klasa zawiera metody pozwalajace na odwolanie sie do pol prywatnych bez nadpisywania ich oraz konstruktory.

5.4 Dokumentacja pliku haszowanie.cpp

```
#include <iostream>
#include "haszowanie.hh"
```

5.5 Dokumentacja pliku haszowanie.hh

plik przechowyjacy deklaracje klasy tablica oraz deklaracje jej pol i metod.

```
#include "dane.hh"
```

Komponenty

- class tablica

  definicja klasy tablica. Definiuje tablice haszujaca(asocjacyjna z mozliwoscia odwolania sie do jej elementu poprzez klucz). Tablica alokowana jest dynamicznie, rozmiar ustalany jest w konstruktorze. Tablica ma narzucony maksymalny rozmiar, poniewaz metoda haszujaca klucze na nim bazuje. Klasa zawiera 2 pola: tab - tablice danych oraz zmienna rozmiar przechowujaca informacje o dlugosci tablicy.

5.6 Dokumentacja pliku losowy_lancuch.hh

plik przechowujacy fukncje do zwracania losowych lancuchow

```
#include <cstdlib>
```

Funkcje

- char LosowyZnak ()

  definicja funkcji LosowyZnak Funkcja inline.
- char * LosowyLancuch ()

  definicja funkcji LosowyLancuch Metoda inline.

5.6.1 Dokumentacja funkcji

5.6.1.1 char* LosowyLancuch ( ) [inline]

Zwraca

losowy lancuch znakow o dlugosci od 0 do 10.

Definicja w linii 25 pliku losowy_lancuch.hh.

5.6.1.2 char LosowyZnak ( ) [inline]

Zwraca

wartosc liczbowa znaku ASCII z przedzialu od A do Z.

Definicja w linii 15 pliku losowy_lancuch.hh.

5.7 Dokumentacja pliku main.cpp

```
#include <iostream>
#include <cstdlib>
#include "losowy_lancuch.hh"
#include "haszowanie.hh"
#include "dane.hh"
#include "benchmark.hh"
#include <ctime>
#include <fstream>
```

Funkcje

- int main ()

5.7.1 Dokumentacja funkcji

5.7.1.1 int main ( )

Definicja w linii 26 pliku main.cpp.

Laboratorium 5 - Sprawozdanie

Wojciech Makuch

15 kwietnia 2015

6 Zadanie

Program framework benchmarkujacy dla zaimplementowanego abstrakcyjnego typu danych: tablica haszujaca.

7 Realizacja

Program wykonano na bazie zwyklej tablicy alokowanej dynamicznie o zadanym maksymalnym rozmiarze. Rozmiar nie może byc zmienny, ponieważ metody haszujace bazują na operacji dzielenia modulo przez ten rozmiar, jego zmiana spowodowałaby znaczne pogorszenie złożoności obliczeniowej. Program zwiera 2 struktury: pierwsza przechowuje dane(element, wartość), druga to tablica przechowujaca te dane. Program zawiera 2 metody haszujace. Pierwsza z nich bazuje na operacji dzielenia modulo przez maksymalny rozmiar wartosci liczbowej kodu ASCII klucza. Druga metoda wykorzystuje wzor q-(dodany wartosci klucza)/q, gdzie q jest polowa maksymalnego rozmiaru i q jest liczba nieparzysta. Zasada wypelniania tablicy jest nastepujaca:

1. WezKlucz
2. i← PierwszeHaszowanie
3. jeśli tab[i]=0, to wypełnij komórke.
   w przypadku przeciwnym:
4. $j \leftarrow$ DrugieHaszowanie
5. jeśli tab[i+j]=0, to wypełnij komórkę.
   w przypadku przeciwnym:
6. Wypelnij najblizsza wolno komorkę.

Ponadto program zawiera przeciążenie operatora [] pozwalającego odnieść się do wybranej komórki tablicy haszującej(asocjacyjnej) za pomocą klucza. Algorytm działa na tym samym schemacie, co wypłenianie.

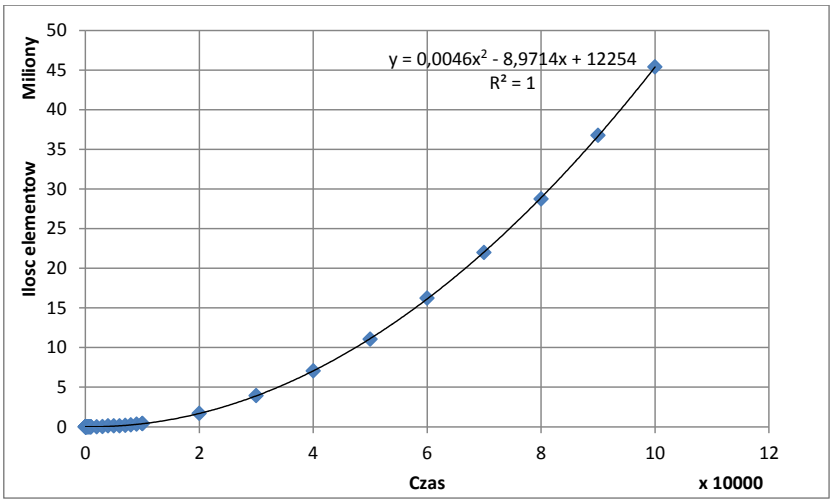
8 Działanie

Program tak samo jak poprzednie nie udostępnia użytkownikowi interfesju. Po uruchomieniu jedynie wyswietla dane, które są jednocześnie zapisywane do pliku o nazwie *pomiar_czas_5.txt*. Główna funkcja programu wywołuje jednynie funkcje *benchmarkuj()*, która zawiera pętle zliczania i zpaisywania uzyskanych wyników.

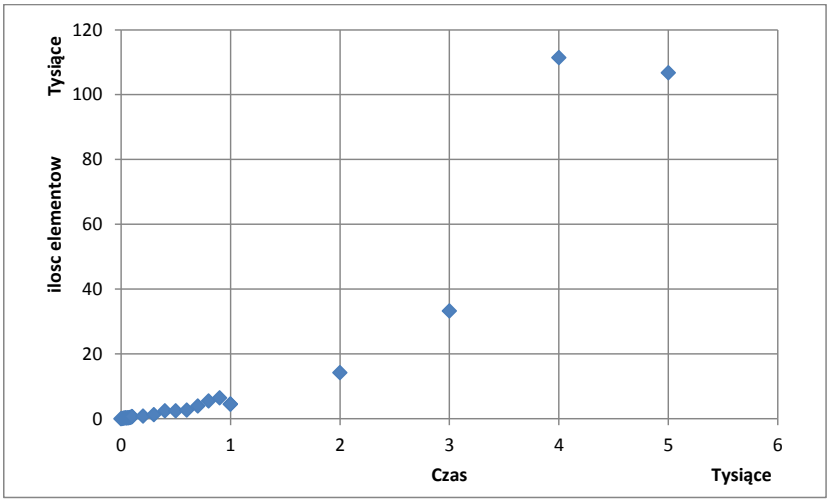
9 Wyniki

Przetestowano czas wypełniania tablicy haszującej elementami losowymi. Uzyskane dane przedstawiono na rys 1. Wynika z nich wprost, że złożoność obliczeniowa wynosi $O(n^2)$, lecz po aproksymacji widać, ze współczynik przy

x^2 jest bardzo mały(0.0046), co oznacza, że wykres mozna również przybliżyć prostą. Wtedy złożoność obliczeniowa wyniesie $O(n)$. jest to najgorszy możliwy przypadek szukania pustej komórki przechodząc się kolejno po elementach tablicy, co dzieje się dosyć często przy wypełnianiu tablicy tak duzą ilocią elementów. Na rys 2. pokazano ten sam wykres dla mneijszej ilosci danych, Widać tu, że żadna funkcja nie jest ani rosnąca, ani malejąca. Może to sugerować złożoność obliczeniową O(1) dla małej ilości elementów.



Rysunek 1: Wykres złożoności obliczeniowej dla dużej ilości elementów.



Rysunek 2: Wykres złożoności obliczeniowej dla małej ilości elementów

10 Podsumowanie

Złozoność obliczeniowa algorytmu wypełniania tablicy haszującej w najgorszym przypadku dla dużej ilości elementów jest $O(n)$ lub $O(n^2)$ natomiast w najlepszym przypadku dla małej ilości jest $O(1)$.

11 Komentarz

Do utworzenia dokumentacji wykorzystano system Doxygen. Funkcja pomiaru czasu dla systemu Windows pobrana ze strony dr. J. Mierzwy. Program skompilowano w środowisku Code::Blocks. Do stworzonia wykresu posłużono się pakietem MS Excel, sprawozdanie napisano używając systemu LaTeX.