Podstawowe struktury danych

1

Wygenerowano przez Doxygen 1.8.6

Śr, 18 mar 2015 15:15:59

Spis treści

1	Spra	wozdai	nie	1
	1.1	Zadan	e	1
	1.2	Wyniki		1
	1.3	Podsu	mowanie	2
2	Inde	ks hiera	archiczny	3
	2.1	Hierard	chia klas	3
3	Inde	ks klas		5
	3.1	Lista k	las	5
4	Inde	ks plike	ów	7
	4.1	Lista p	lików	7
5	Dok	umenta	cja klas	9
	5.1	Dokum	nentacja klasy benchmark	9
		5.1.1	Opis szczegółowy	9
		5.1.2	Dokumentacja funkcji składowych	9
			5.1.2.1 analyze	10
			5.1.2.2 test	10
	5.2	Dokum	nentacja klasy list	10
		5.2.1	Opis szczegółowy	12
		5.2.2	Dokumentacja konstruktora i destruktora	12
			5.2.2.1 list	12
			5.2.2.2 ~list	12
		5.2.3	Dokumentacja funkcji składowych	12
			5.2.3.1 pop	12
			5.2.3.2 push	13
			5.2.3.3 size	13
			5.2.3.4 test	13
		5.2.4	Dokumentacja atrybutów składowych	14
			5.2.4.1 head	14
	53	Dokum	nentacia struktury listrinode	14

iv SPIS TREŚCI

		5.3.1	Opis szczegółowy	14
		5.3.2	Dokumentacja konstruktora i destruktora	15
			5.3.2.1 node	15
		5.3.3	Dokumentacja atrybutów składowych	15
			5.3.3.1 data	15
			5.3.3.2 next	15
6	Dok	umenta	cja plików	17
	6.1	Dokum	nentacja pliku benchmark.cpp	17
	6.2	Dokum	nentacja pliku benchmark.hh	17
	6.3	Dokum	nentacja pliku generator.cpp	18
		6.3.1	Dokumentacja funkcji	19
			6.3.1.1 data_generator	19
	6.4	Dokum	nentacja pliku generator.hh	19
		6.4.1	Dokumentacja funkcji	20
			6.4.1.1 data_generator	20
	6.5	Dokum	nentacja pliku list.cpp	20
	6.6	Dokum	nentacja pliku list.hh	21
	6.7	Dokum	nentacja pliku main.cpp	22
		6.7.1	Dokumentacja funkcji	22
			6.7.1.1 main	22
	6.8	Dokum	nentacja pliku strona.dox	23
Inc	deks			24

Sprawozdanie

Data

11.03.2015r.

Wersja

0.1

1.1 Zadanie

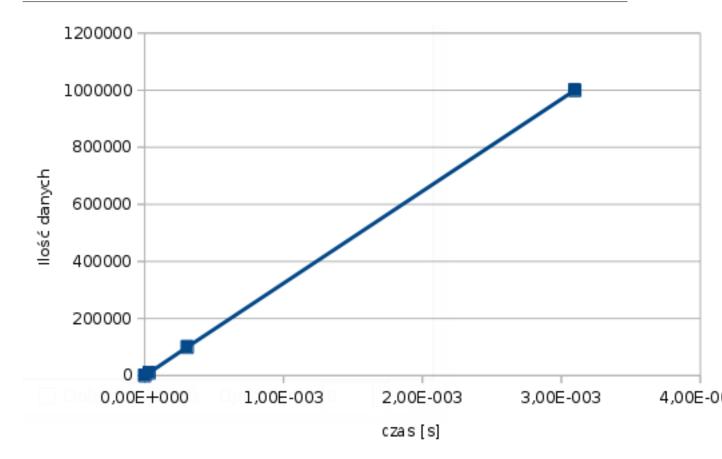
Celem ćwiczenia było stworzenie programu benchmarkującego , który dla wybranych danych będzie zliczał średni czas wykonania dowolnego algorytmu (w tym przypadku mnożenia elementów tablicy przez 2). Należało również stworzyć program generujący losowe liczby.

1.2 Wyniki

Dla dziesięciu milionów liczb program zwraca 7 danych wyjściowych (zgodnie z algorytmem 10^n , gdzie n jest równocześnie ilością zwracanych czasów oraz maxymalną liczbą danych dla jakiej przeprowadzany był test

Na podstawie otrzymanych danych mamy :

2 Sprawozdanie



1.3 Podsumowanie

Wykres dodany do dokumentacji z niewiadomych względów nie jest wyświetlany poprawnie (dodano sprawozdanie również w formacie pdf). Zgodnie z przewidywaniami złożoność obliczeniowa jest liniowa , jedyną rzeczą która zwraca uwagę jest fakt iż czas wykonania jednej operacji jest dłuższy od czasu wykonania 10 operacji. Dla wiekszej ilości danych wyniki są poprawne Wydaje się , że zbyt mało danych jest obecnych w środkowej częsci wykresu co powinno być zostać zmienione w celu poprawy jakości odbioru wykresu (dla charakterystyki liniowej jest to akurat bez znaczenia ale .np dla logarytmicznej było by widoczne).

Indeks hierarchiczny

2.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:		
benchmark	 	9
list	 	. 10
list::node		4.

4	Indeks hierarchiczny

Indeks klas

3.1 Lista klas

-						1 / . 1	
Lutai znaidilla	CIA MIACV	ctrulzturv	TIMIA LIMI	Ortolov Wroz	7 100	krotkimi	anicami:
TUTAL ZHARUMA	SIE NIASV.	SHUNIUI V.		elleisv wiaz	/ 10.11	NIOINIIII	oonsann.
Tutaj znajdują	0.6,	o o , ,				•	00.00

benchmark		 				 												 						,	3
list		 				 												 						10)
list::node		 				 						 						 						14	4

6 Indeks klas

Indeks plików

4.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

enchmark.cpp	
Deklaracja funkzji z klasy Benchmark	. 17
enchmark.hh	
Definicja klasy Benchmark	. 17
enerator.cpp	
Deklaracja funckji generujacej liczby losowe	. 18
enerator.hh	
Definicja generatora liczb losowych	. 19
st.cpp	
Deklaracja klasy list	. 20
st.hh	
Definicja klasy list	. 21
ain.cpp	. 22

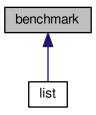
Indeks plików 8

Dokumentacja klas

5.1 Dokumentacja klasy benchmark

#include <benchmark.hh>

Diagram dziedziczenia dla benchmark



Metody publiczne

• void analyze (int repeat, int data_amount)

Metoda analyze zlicza czas wykonywania funkcji test() Uwaga! do poprawnego działania wymagane jest posiadanie programu gnuplot.

Metody prywatne

virtual void test (int length)=0
 Metoda test funkcja wirtualna, której czas działania ma być aproksymowany przez metoda analyze()

5.1.1 Opis szczegółowy

Definicja w linii 11 pliku benchmark.hh.

5.1.2 Dokumentacja funkcji składowych

10 Dokumentacja klas

5.1.2.1 void benchmark::analyze (int repeat, int data_amount)

Metoda analyze zlicza czas fukcji test()

Przykład wywołania funkcji:

analyze(100,7) -> Przeprowadza analize czesu trwania funkcji test() dla 1 miliona danych , każdy czas trwania funkcji jest ustalany na podstawie średniej arytmetycznej ze 100 prób.

Parametry

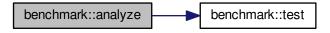
	in	repeat	- ilość powtórzeń testu
ſ	in	data_amount	- ilosc wynikowych danych podawana jako potega liczby 10

Zwraca

plik.dat z czasami poszczegolnych pomiarow oraz ilosc testowanych danych oraz plik plot.png bedacy graficznym przedstawieniem danych na wykresie

Definicja w linii 15 pliku benchmark.cpp.

Oto graf wywołań dla tej funkcji:



5.1.2.2 virtual void benchmark::test (int length) [private], [pure virtual]

Implementowany w list.

Oto graf wywoływań tej funkcji:



Dokumentacja dla tej klasy została wygenerowana z plików:

- · benchmark.hh
- · benchmark.cpp

5.2 Dokumentacja klasy list

#include <list.hh>

Diagram dziedziczenia dla list

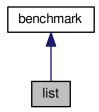
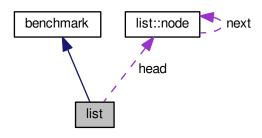


Diagram współpracy dla list:



Komponenty

struct node

Metody publiczne

void push (int insert)

Metoda push() definiuje dodawanie elementu do początku listy.

• void pop ()

Metoda pop() definiuje usuwanie elementu z początku listy.

• unsigned size ()

Metoda size() zwraca ilość elementów listy.

• list ()

Konstruktor inicjalizujący zmienną wskaźnikową , która domyślnie ma pokazywać na NULL.

∼list ()

Destruktor usuwa wszystkie elementy z listy za pomocą funkcji pop.

void test (int length)

Metoda test() realizuje operacje zapelniania listy ustalonymi danymi, czas będzie zliczany <<<<DOKO Ńcz>="">>>>> 12 Dokumentacja klas

Atrybuty publiczne

node * head

Pole będące pierwszym wskaźnikiem na elementy listy.

5.2.1 Opis szczegółowy

Definicja w linii 17 pliku list.hh.

5.2.2 Dokumentacja konstruktora i destruktora

5.2.2.1 list::list()

Definicja w linii 9 pliku list.cpp.

5.2.2.2 list::∼list ()

Definicja w linii 13 pliku list.cpp.

Oto graf wywołań dla tej funkcji:



5.2.3 Dokumentacja funkcji składowych

5.2.3.1 void list::pop ()

Metoda pop() usuwa z listy ostatni element lub zwraca komunikat o błędzie w przypadku próby usnięcia elementu z pustej listy.

Definicja w linii 42 pliku list.cpp.

Oto graf wywoływań tej funkcji:



5.2.3.2 void list::push (int insert)

Metoda push() wczytuje liczbę naturalną na listę

Przykład wywołania funkcji:

push(10) - Na początek listy zostanie wprowadzona liczba 10.

Parametry

in	insert	- dodawany element

Definicja w linii 24 pliku list.cpp.

Oto graf wywoływań tej funkcji:



5.2.3.3 unsigned list::size ()

Metoda size() zwraca ilość elementów znajdujących się na liście.

Definicja w linii 57 pliku list.cpp.

Oto graf wywoływań tej funkcji:



5.2.3.4 void list::test (int length) [inline], [virtual]

Implementuje benchmark.

Definicja w linii 62 pliku list.hh.

14 Dokumentacja klas

Oto graf wywołań dla tej funkcji:



5.2.4 Dokumentacja atrybutów składowych

5.2.4.1 node* list::head

Definicja w linii 35 pliku list.hh.

Dokumentacja dla tej klasy została wygenerowana z plików:

- · list.hh
- · list.cpp

5.3 Dokumentacja struktury list::node

Diagram współpracy dla list::node:



Metody publiczne

• node ()

Atrybuty publiczne

- int data
- node * next

5.3.1 Opis szczegółowy

Definicja w linii 20 pliku list.hh.

5.3.2 Dokumentacja konstruktora i destruktora

5.3.2.1 list::node::node() [inline]

Definicja w linii 24 pliku list.hh.

5.3.3 Dokumentacja atrybutów składowych

5.3.3.1 int list::node::data

Definicja w linii 22 pliku list.hh.

5.3.3.2 node* list::node::next

Definicja w linii 23 pliku list.hh.

Dokumentacja dla tej struktury została wygenerowana z pliku:

· list.hh

16 Dokumentacja klas

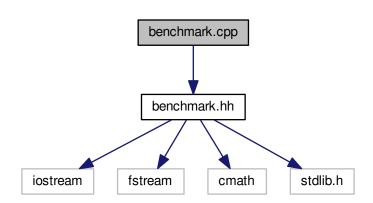
Dokumentacja plików

6.1 Dokumentacja pliku benchmark.cpp

Deklaracja funkzji z klasy Benchmark.

#include "benchmark.hh"

Wykres zależności załączania dla benchmark.cpp:



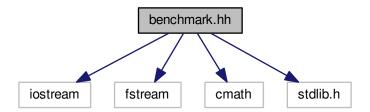
6.2 Dokumentacja pliku benchmark.hh

Definicja klasy Benchmark.

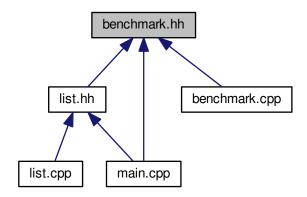
```
#include <iostream>
#include <fstream>
#include <cmath>
#include "stdlib.h"
```

18 Dokumentacja plików

Wykres zależności załączania dla benchmark.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

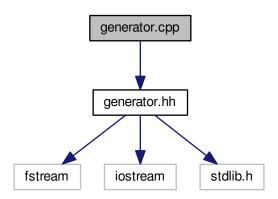
· class benchmark

6.3 Dokumentacja pliku generator.cpp

Deklaracja funckji generujacej liczby losowe.

#include "generator.hh"

Wykres zależności załączania dla generator.cpp:



Funkcje

• bool data_generator (int data_amount)

Generuje liczby losowe.

6.3.1 Dokumentacja funkcji

6.3.1.1 bool data_generator (int data_amount)

Funkcja generuje naturalne liczby losowe z przedziału 0-100, ktore nastepnie sa zapisywane do pliku random_-data.dat

Parametry

in	data_amount	- ilosc liczb wynikowych ktore chcemy uzyskac

Definicja w linii 9 pliku generator.cpp.

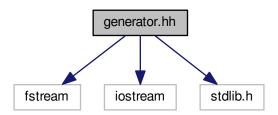
6.4 Dokumentacja pliku generator.hh

Definicja generatora liczb losowych.

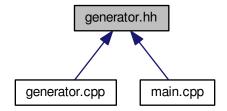
```
#include <fstream>
#include <iostream>
#include <stdlib.h>
```

20 Dokumentacja plików

Wykres zależności załączania dla generator.hh:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

bool data_generator (int data_amount)
 Generuje liczby losowe.

6.4.1 Dokumentacja funkcji

6.4.1.1 bool data_generator (int data_amount)

Funkcja generuje naturalne liczby losowe z przedziału 0-100, ktore nastepnie sa zapisywane do pliku random_-data.dat

Parametry

in	data_amount	- ilosc liczb wynikowych ktore chcemy uzyskac

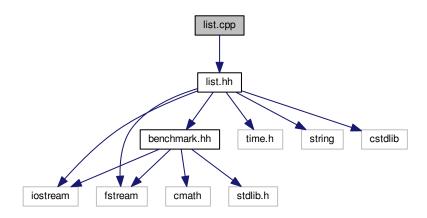
Definicja w linii 9 pliku generator.cpp.

6.5 Dokumentacja pliku list.cpp

Deklaracja klasy list.

#include "list.hh"

Wykres zależności załączania dla list.cpp:

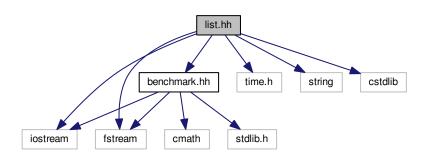


6.6 Dokumentacja pliku list.hh

Definicja klasy list.

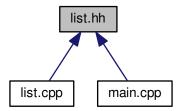
#include <iostream>
#include <time.h>
#include <string>
#include <fstream>
#include <cstdlib>
#include "benchmark.hh"

Wykres zależności załączania dla list.hh:



22 Dokumentacja plików

Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



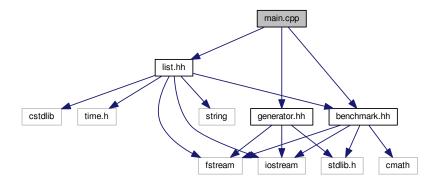
Komponenty

- class list
- struct list::node

6.7 Dokumentacja pliku main.cpp

```
#include "list.hh"
#include "benchmark.hh"
#include "generator.hh"
```

Wykres zależności załączania dla main.cpp:



Funkcje

• int main ()

6.7.1 Dokumentacja funkcji

6.7.1.1 int main ()

Definicja w linii 6 pliku main.cpp.

Oto graf wywołań dla tej funkcji:



6.8 Dokumentacja pliku strona.dox

Skorowidz

list, 12
analyze benchmark, 9
benchmark, 9 analyze, 9 test, 10
benchmark.cpp, 17 benchmark.hh, 17
data list::node, 15 data_generator generator.cpp, 19 generator.hh, 20
generator.cpp, 18 data_generator, 19 generator.hh, 19 data_generator, 20
head list, 14
list, 10 ∼list, 12
head, 14 list, 12 pop, 12 push, 12 size, 13 test, 13 list.cpp, 20 list.hh, 21 list::node, 14 data, 15 next, 15 node, 15
list, 12 pop, 12 push, 12 size, 13 test, 13 list.cpp, 20 list.hh, 21 list::node, 14 data, 15 next, 15
list, 12 pop, 12 push, 12 size, 13 test, 13 list.cpp, 20 list.hh, 21 list::node, 14 data, 15 next, 15 node, 15 main main.cpp, 22 main.cpp, 22

pop

list, 12
push
list, 12
size
list, 13
strona.dox, 23
test
benchmark, 10
list, 13