

Sortowanie

1.5

Generated by Doxygen 1.8.4

Tue Apr 1 2014 18:39:27

Contents

1	Sortowanie	1
1.1	Najważniejsze cechy	1
1.2	Opis algorytmu	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	benchmark Class Reference	7
4.1.1	Detailed Description	8
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	benchmark	8
4.1.3	Member Function Documentation	8
4.1.3.1	algorytm	8
4.1.3.2	czas_start	9
4.1.3.3	czas_stop	9
4.1.3.4	ile_czasu	10
4.1.3.5	wykonaj_algorytm_sortowanie	10
4.1.3.6	wyswietl_wszystko	11
4.1.4	Member Data Documentation	11
4.1.4.1	czas	11
4.1.4.2	czas_caly	12
4.1.4.3	elapsedTime	12
4.1.4.4	t1	12
4.1.4.5	t2	12
4.2	stos_tablica Class Reference	12
4.2.1	Detailed Description	13
4.2.2	Constructor & Destructor Documentation	14
4.2.2.1	stos_tablica	14

4.2.2.2	~stos_tablica	14
4.2.3	Member Function Documentation	14
4.2.3.1	build_heap	14
4.2.3.2	disassemble_heap	14
4.2.3.3	get_lewy	14
4.2.3.4	get_prawy	14
4.2.3.5	get_rozmiar	15
4.2.3.6	heap_sort	15
4.2.3.7	merge_sort	15
4.2.3.8	mergesort	16
4.2.3.9	pokaz_elementy	16
4.2.3.10	przypisz	16
4.2.3.11	quick_sort	17
4.2.3.12	quicksort	17
4.2.3.13	swap	17
4.2.3.14	wczytaj_dane	17
4.2.3.15	wczytaj_dane	18
4.2.3.16	wypelnij_losowo	18
4.2.3.17	wyzeruj_stos	18
4.2.3.18	wyzeruj_stos_caly	18
4.2.4	Member Data Documentation	19
4.2.4.1	dane	19
4.2.4.2	danetmp	19
4.2.4.3	heap_size	19
4.2.4.4	lewy	19
4.2.4.5	prawy	19
4.2.4.6	rozmiar	19
4.2.4.7	rozmiar_tmp	19
5	File Documentation	21
5.1	/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/doc/pages/strona.dox File Reference	21
5.2	/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/inc/benchmark.hh File Reference	21
5.2.1	Detailed Description	22
5.3	/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/inc/stos_tablica.hh File Reference	22
5.3.1	Detailed Description	23
5.4	/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/benchmark.cpp File Reference	24
5.4.1	Detailed Description	24
5.5	/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/main.cpp File Reference	24
5.5.1	Detailed Description	25

5.5.2	Function Documentation	25
5.5.2.1	main	25
5.6	/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/stos_tablica.cpp File Reference	25
 Index		 27

Chapter 1

Sortowanie

Author

Pawel Zurek

Date

22.03.2014

Version

1.5

Program umożliwia liczenie czasu trwania operacji sortowania danych

1.1 Najważniejsze cechy

Program służy do liczenia czasu trwania operacji sortowania:

- > Quick Sort (Sortowanie szybkie)
- > Merge Sort (Sortowanie przez scalanie)
- > Heap Sort (Sortowanie przez budowanie kopca)

1.2 Opis algorytmu

Algorytm w tym zadaniu to 10 petli :

- > posortowanie stosu za pomocą Quick Sort
- > posortowanie stosu za pomocą Merge Sort
- > posortowanie stosu za pomocą Heap Sort

,na którą składa się wypełnienie następującą ilością elementów:

- > 10
- > 100
- > 1000
- > 10000
- > 100000

-> 1000000

Wszystkie posortowania zostały przeprowadzone dla elementów uporządkowanych rosnąco, malejąco oraz losowo. Informacje te zostały zawarte na wykresach dostępnych w pliku głównym programu.

Czasy każdego sortowania dla każdej ilości elementów razem z czasem całkowitym są wyprowadzane na standardowe wyjście.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

benchmark	Modeluje pojecie Benchmark	7
stos_tablica	Modeluje pojecie Stos	12

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/inc/ benchmark.hh	
Definicje funkcji dla klasy benchmark	21
/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/inc/ stos_tablica.hh	
Definicje funkcji dla klasy Stos zdefiniowanej tablica	22
/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/ benchmark.cpp	
Plik zawiera funkcje z klasy benchmark	24
/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/ main.cpp	
Plik zawiera funkcje main()	24
/home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/ stos_tablica.cpp	
.	25

Chapter 4

Class Documentation

4.1 benchmark Class Reference

Modeluje pojecie Benchmark.

```
#include <benchmark.hh>
```

Public Member Functions

- `benchmark ()`
Konstruktor klasy Benchmark.
- `void czas_start ()`
Funkcja pomocnicza mierzenia czasu.
- `void czas_stop ()`
Funkcja pomocnicza mierzenia czasu.
- `double ile_czasu ()`
Funkcja obliczania czasu dzialania programu.
- `void wykonaj_algorytm_sortowanie ()`
Funkcja wykonujaca algorytm.
- `void algorytm ()`
Funkcja wykonujaca algorytm.
- `void wyswietl_wszystko (double *c, double *q, double *m, double *h, int n, int s)`
Funkcja wyswietlania wynikow.

Private Attributes

- `double elapsedTime`
Pole typu double, bedzie uzywane do mierzenia czasu dzialania pojedynczego wypelniania.
- `double czas`
Pole typu double, bedzie uzywane do mierzenia calkowitego czasu dzialania programu.
- `double czas_caly`
Pole typu double, bedzie uzywane do mierzenia calkowitego czasu dzialania programu. !
- `timeval t1`
Pole typu timeval, pomoc do liczenia czasu dzialania operacji krotkich (tzn pojedynczego dzialania)
- `timeval t2`

4.1.1 Detailed Description

Modeluje pojecie Benchmark.

Klasa sluzy do przeprowadzenia Benchmarku programu, tzn : -> wczytania dowolnego zestawu danych o ilosci elementow :

-> 10

-> 100

-> 1000

-> 10000

-> 100000

-> 1000000

Trzema roznymi metodami :

-> Quick Sort

-> Merge Sort

-> Heap Sort

Można jeszcze wybrać ile razy ma zostać wykonany program.

-> Na koniec zostana wyswietlone czasy kazdej akcji z osobna oraz czas calkowity

Definition at line 41 of file benchmark.hh.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 `benchmark::benchmark()` `[inline]`

Konstruktor klasy Benchmark.

Konstruktor jest bezparametryczny, inicjalizuje wszystkie skladowe klasy wartosciami zerowymi.

Definition at line 71 of file benchmark.hh.

4.1.3 Member Function Documentation

4.1.3.1 `void benchmark::algorytm()`

Funkcja wykonujaca algorytm.

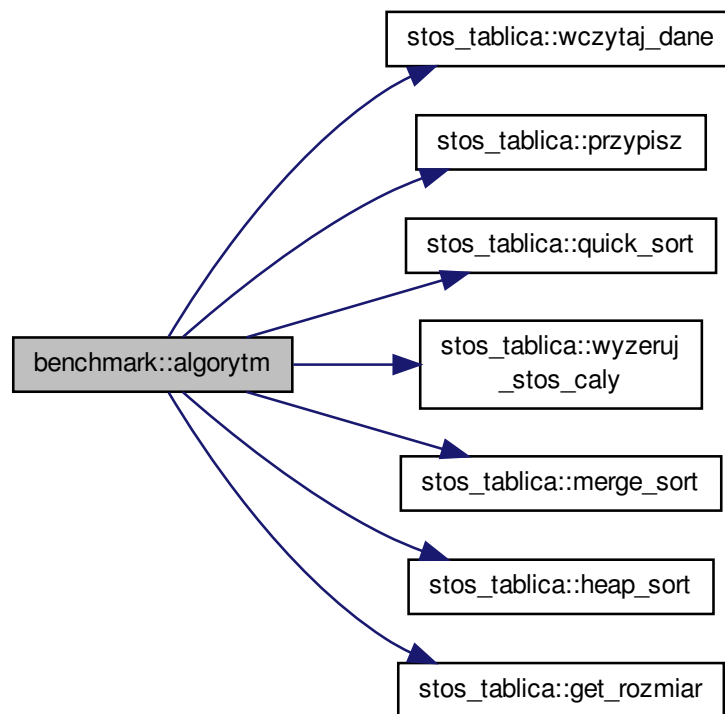
Wykonanie algorytmu ma przebieg :

-> posortowanie wybranego pliku trzema metodami:

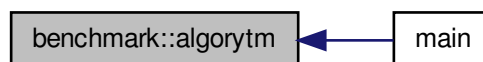
- Quick Sort
- Merge Sort
- Heap Sort -> po kazdym posortowaniu, obiekt jest calkowicie zerowany
- > wszystko to wykonuje sie zadana przez uzytkownika ilosc razy

Definition at line 98 of file benchmark.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.3.2 void benchmark::czas_start ()

Funkcja pomocnicza mierzenia czasu.

Funkcja zaczyna liczyć czas od momentu wywołania tej metody. Służy do liczenia czasu wykonywania pojedynczego wypełniania stosu/kolejki.

Definition at line 14 of file benchmark.cpp.

4.1.3.3 void benchmark::czas_stop ()

Funkcja pomocnicza mierzenia czasu.

Funkcja konczy liczyc czas od momentu wywolania tej metody Sluzy do liczenia czasu wykonywania pojedynczego wypelniania stosu/kolejki

Definition at line 17 of file benchmark.cpp.

4.1.3.4 double benchmark::ile_czasu ()

Funkcja obliczania czasu dzialania programu.

Funkcja podaje czas wykonywania pojedynczego wypelniania stosu/kolejki

Returns

elapsedTime -> zmienna typu double (wynik obliczen)

Definition at line 20 of file benchmark.cpp.

4.1.3.5 void benchmark::wykonaj_algorytm_sortowanie ()

Funkcja wykonujaca algorytm.

Wykonanie algorytmu ma przebieg :

-> posortowanie danych metoda Quick Sort

-> posortowanie danych metoda Merge Sort

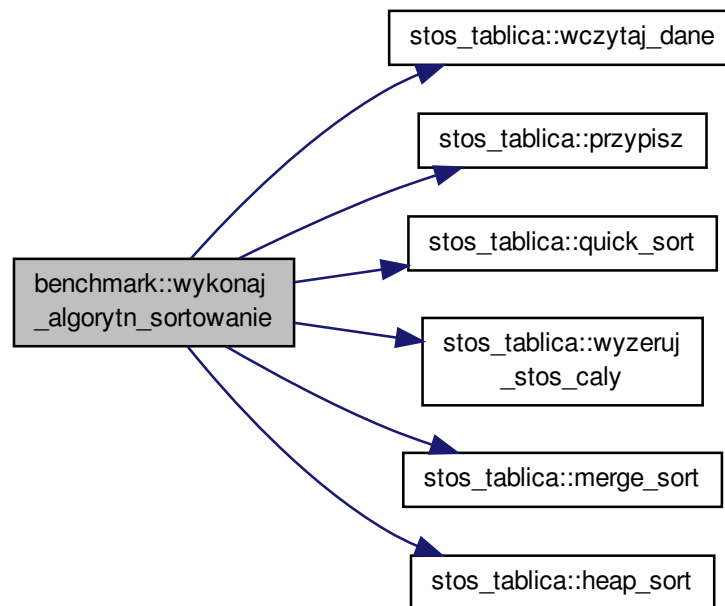
-> posortowanie danych metoda Heap Sort

Dla :

- 10 elementow
- 100 elementow
- 1000 elementow
- 10000 elementow
- 100000 elementow
- 1000000 elementow

Definition at line 27 of file benchmark.cpp.

Here is the call graph for this function:



4.1.3.6 void benchmark::wyswietl_wszystko (double * c, double * q, double * m, double * h, int n, int s)

Funkcja wyswietla wyniki.

Funkcja wyswietla wszystkie czasy liczone w programie. Tzn: Czasy wszystkich wykonan dla kazdego sortowania, najwolniejsze, najszybsze oraz srednie wykonanie. Dodatkowo czas wykonania calej petli za kazdym razem

Parameters

<i>c</i>	-> wskaznik na zmienna typu double, przechowuje adres pola czasu calkowitego (czasy)
<i>q</i>	-> wskaznik na zmienna typu double, przechowuje adres pola czasu wykonywania Quick Sort (quick)
<i>m</i>	-> wskaznik na zmienna typu double, przechowuje adres pola czasu wykonywania Merge Sort (merge)
<i>h</i>	-> wskaznik na zmienna typu double, przechowuje adres pola czasu wykonywania Heap Sort (heap)
<i>n</i>	-> zmienna typu int, przechowuje adres pola, w ktorym jest informacja o tym ile razy zostal wykonany algorytm (ile_razy)
<i>s</i>	-> zmienna typu int, przechowuje adres pola rozmiaru tablicy (size)

Definition at line 149 of file benchmark.cpp.

4.1.4 Member Data Documentation

4.1.4.1 double benchmark::czas [private]

Pole typu double, bedzie uzywane do mierzenia calkowitego czasu dzialania programu.

Definition at line 51 of file benchmark.hh.

4.1.4.2 double benchmark::czas_caly [private]

Pole typu double, bedzie uzywane do mierzenia calkowitego czasu dzialania programu. !

Definition at line 55 of file benchmark.hh.

4.1.4.3 double benchmark::elapsedTime [private]

Pole typu double, bedzie uzywane do mierzenia czasu dzialania pojedynczego wypelniania.

Definition at line 47 of file benchmark.hh.

4.1.4.4 timeval benchmark::t1 [private]

Pole typu timeval, pomoc do liczenia czasu dzialania operacji krotkich (tzn pojedynczego dzialania)

Definition at line 59 of file benchmark.hh.

4.1.4.5 timeval benchmark::t2 [private]

Definition at line 59 of file benchmark.hh.

The documentation for this class was generated from the following files:

- /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/inc/[benchmark.hh](#)
- /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/[benchmark.cpp](#)

4.2 stos_tablica Class Reference

Modeluje pojecie Stos.

```
#include <stos_tablica.hh>
```

Public Member Functions

- [stos_tablica](#) ()
Konstruktor klasy stos.
- [~stos_tablica](#) ()
Destruktor klasy stos.
- void [wczytaj_dane](#) ()
Funkcja wczytywania.
- void [wczytaj_dane](#) (string nazwa)
Funkcja wczytywania.
- void [pokaz_elementy](#) ()
Funkcja wyswietlajaca.
- int [get_rozmiar](#) ()
Funkcja sprawdzania rozmiar stosu.
- void [quicksort](#) (int left, int right)
Funkcja sortujaca.
- void [quick_sort](#) ()
Funkcja wywolujaca funkcje sorujaca Quick Sort.
- void [swap](#) (int *x, int *y)
Funkcja zamieniania wartosci.

- void [wypelnij_losowo](#) ()
Funkcja wypelniajaca stos losowymi wartosciami.
- void [wyzeruj_stos](#) ()
Funkcja zerujaca stos.
- void [wyzeruj_stos_caly](#) ()
Funkcja zerujaca caly stos.
- int [get_lewy](#) ()
Funkcja dostepu do prywatnego pola klasy lewy.
- int [get_prawy](#) ()
Funkcja dostepu do prywatnego pola klasy prawy.
- void [mergesort](#) (int pocz, int kon)
Funkcja sortujaca.
- void [merge_sort](#) ()
Funkcja wywolujaca funkcje sorujaca Merge Sort.
- void [build_heap](#) ()
Funkcja budujaca kopiec (Heap).
- void [disassemble_heap](#) ()
Funkcja rozbierajaca kopiec (Heap).
- void [heap_sort](#) ()
Funkcja wywolujaca funkcje sorujaca Heap Sort.
- void [przypisz](#) ()
Funkcja przypisujaca.

Private Attributes

- int * [dane](#)
Pole typu int, bedzie uzywane jako stosu z danymi.
- int * [danetmp](#)
Pole typu int, bedzie uzywane jako stosu z danymi sprawdzajacymi, pomocniczymi.
- int [rozmiar](#)
Pole typu int, bedzie uzywane jako rozmiar tabeli.
- int [rozmiar_tmp](#)
Pole typu int, bedzie uzywane jako pomocnicza wartosc jako rozmiar stosu.
- int [lewy](#)
Pole typu int, bedzie uzywane jako pomocnicza wartosc do metody Quick Sort.
- int [prawy](#)
Pole typu int, bedzie uzywane jako pomocnicza wartosc do metody Quick Sort.
- int [heap_size](#)
Pole typu int, bedzie uzywane jako pomocnicza wartosc do metody Heap Size.

4.2.1 Detailed Description

Modeluje pojecie Stos.

Stos jest klasa zawierajaca dynamicznie zaalokowane 2 tablice Pierwsza z nich to tablica z danymi, na ktorych beda wykonywane operacje. Druga z nich sluzi jako tablica do przechowywania tymczasowych danych.

Definition at line 34 of file stos_tablica.hh.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 `stos_tablica::stos_tablica ()` `[inline]`

Konstruktor klasy `stos`.

Konstruktor jest bezparametryczny, inicjalizuje wszystkie składowe klasy wartościami zerowymi.

Definition at line 74 of file `stos_tablica.hh`.

4.2.2.2 `stos_tablica::~~stos_tablica ()` `[inline]`

Destruktor klasy `stos`.

Usuwa dynamicznie zaalokowaną tablicę

Definition at line 81 of file `stos_tablica.hh`.

4.2.3 Member Function Documentation

4.2.3.1 `void stos_tablica::build_heap ()`

Funkcja budująca kopiec (`Heap`).

Funkcja tworzy kopiec z wstępnie posortowanymi danymi. Reszta zostanie posortowana przy jego rozbieraniu.

Definition at line 186 of file `stos_tablica.cpp`.

4.2.3.2 `void stos_tablica::disassemble_heap ()`

Funkcja rozbierająca kopiec (`Heap`).

Funkcja rozbiera kopiec, jednocześnie sortując dane.

Definition at line 205 of file `stos_tablica.cpp`.

4.2.3.3 `int stos_tablica::get_lewy ()` `[inline]`

Funkcja dostępu do prywatnego pola klasy `lewy`.

Returns

`lewy` -> obiekt typu `int`, indeks pierwszego elementu stosu (zazwyczaj 0)

Definition at line 186 of file `stos_tablica.hh`.

4.2.3.4 `int stos_tablica::get_prawy ()` `[inline]`

Funkcja dostępu do prywatnego pola klasy `prawy`.

Returns

`prawy` -> obiekt typu `int`, indeks ostatniego elementu stosu (zazwyczaj równe rozmiarowi)

Definition at line 193 of file `stos_tablica.hh`.

4.2.3.5 int stos_tablica::get_rozmiar () [inline]

Funkcja sprawdzania rozmiar stosu.

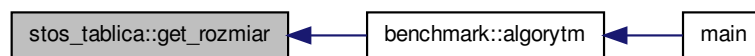
Funkcja podaje aktualny rozmiar stosu

Returns

rozmiar -> rozmiar stosu

Definition at line 128 of file stos_tablica.hh.

Here is the caller graph for this function:



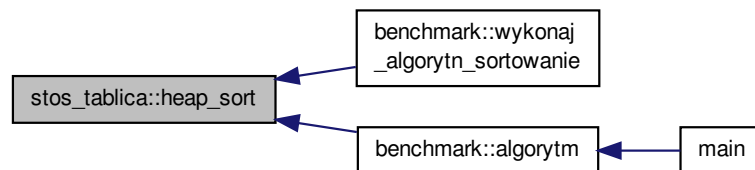
4.2.3.6 void stos_tablica::heap_sort () [inline]

Funkcja wywołująca funkcje sortująca Heap Sort.

Funkcja jest tylko po to, aby można było w klasie innej niż ta uruchomić funkcje sortowania jednym poleceniem

Definition at line 234 of file stos_tablica.hh.

Here is the caller graph for this function:



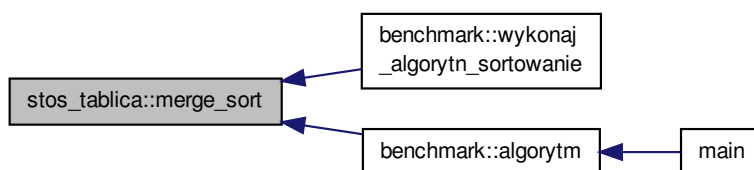
4.2.3.7 void stos_tablica::merge_sort () [inline]

Funkcja wywołująca funkcje sortująca Merge Sort.

Funkcja jest tylko po to, aby można było w klasie innej niż ta uruchomić funkcje sortowania bezparametrycznie

Definition at line 211 of file stos_tablica.hh.

Here is the caller graph for this function:



4.2.3.8 void stos_tablica::mergesort (int pocz, int kon)

Funkcja sortujaca.

Funkcja sortuje dane za pomoca algorytmu Merge Sort

Parameters

<i>pocz</i>	-> Pole typu int, zawiera informacje indeksie poczatkowym sortowanego zbioru
<i>kon</i>	-> Pole typu int, zawiera informacje indeksie koncowym sortowanego zbioru

Definition at line 162 of file stos_tablica.cpp.

4.2.3.9 void stos_tablica::pokaz_elementy ()

Funkcja wyswietlajaca.

Funkcja wyswietla aktualny stan stosu

Definition at line 7 of file stos_tablica.cpp.

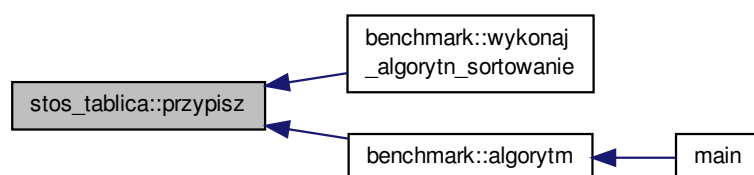
4.2.3.10 void stos_tablica::przypisz ()

Funkcja przypisujaca.

Funkcja przypisuje wartosci dynamicznej, pomocniczej tabeli danetmp do tabeli glownej programu -> dane

Definition at line 66 of file stos_tablica.cpp.

Here is the caller graph for this function:



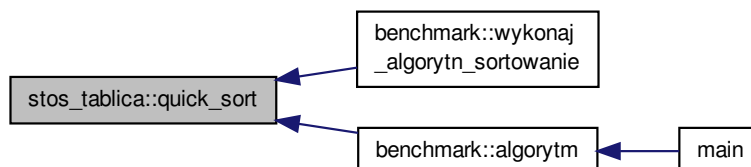
4.2.3.11 void stos_tablica::quick_sort () [inline]

Funkcja wywołująca funkcję sortującą Quick Sort.

Funkcja jest tylko po to, aby można było w klasie innej niż ta uruchomić funkcję sortowania bezparametrycznie

Definition at line 146 of file stos_tablica.hh.

Here is the caller graph for this function:



4.2.3.12 void stos_tablica::quicksort (int left, int right)

Funkcja sortująca.

Funkcja sortuje dane za pomocą algorytmu Quick Sort

Parameters

<i>left</i>	-> Pole typu int, zawiera informacje indeksie początkowym sortowanego zbioru
<i>right</i>	-> Pole typu int, zawiera informacje indeksie końcowym sortowanego zbioru

Definition at line 128 of file stos_tablica.cpp.

4.2.3.13 void stos_tablica::swap (int * x, int * y)

Funkcja zamieniania wartości.

Funkcja zamienia wartościami obiekty, które są argumentami funkcji

Parameters

<i>x</i>	-> Pole typu *int, zawiera adres obiektu, którego wartość ma być zamieniona z obiektem y
<i>y</i>	-> Pole typu *int, zawiera adres obiektu, którego wartość ma być zamieniona z obiektem x

Definition at line 155 of file stos_tablica.cpp.

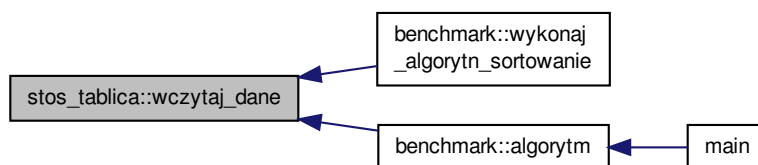
4.2.3.14 void stos_tablica::wczytaj_dane ()

Funkcja wczytywania.

Funkcja wczytuje wartości do tabeli z podanego pliku przez użytkownika.

Definition at line 19 of file stos_tablica.cpp.

Here is the caller graph for this function:



4.2.3.15 void stos_tablica::wczytaj_dane (string nazwa)

Funkcja wczytywania.

Funkcja wczytuje wartosci do tabeli po przez wpisanie nazwy jako argument metody Wykorzystuje metode push jako funkcje wpisujaca do stosu

Parameters

<i>nazwa</i>	-> zmienna typu string, przechowuje nazwe otwieranego pliku
--------------	-------------------------------------------------------------

Definition at line 43 of file stos_tablica.cpp.

4.2.3.16 void stos_tablica::wypelnij_losowo ()

Funkcja wypelniajaca stos losowymi wartosciami.

Funkcja wypelnia stos losowymi wartosciami. Uzytkownik wybiera sam ile elementow ma zostac wpisanych

Definition at line 81 of file stos_tablica.cpp.

4.2.3.17 void stos_tablica::wyzeruj_stos ()

Funkcja zerujaca stos.

Funkcja kasuje dynamicznie zaalokowana tablice dane, zeby mozna bylo wpisac kolejna tablice z tymi samymi wartosciami Dodatkowo zeruje pola klasy zwiazane z ową tablica

Definition at line 101 of file stos_tablica.cpp.

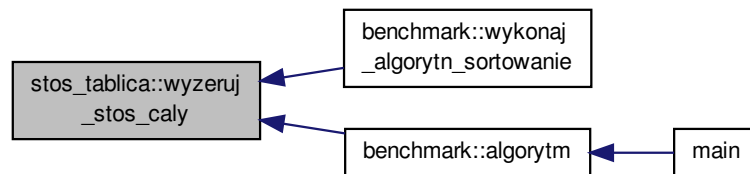
4.2.3.18 void stos_tablica::wyzeruj_stos_caly ()

Funkcja zerujaca caly stos.

Funkcja kasuje wszystkie dynamicznie zaalokowane obiekty. Dodatkowo zeruje wszystkie pola klasy [stos_tablica](#)

Definition at line 112 of file stos_tablica.cpp.

Here is the caller graph for this function:



4.2.4 Member Data Documentation

4.2.4.1 `int* stos_tablica::dane` `[private]`

Pole typu `int`, bedzie uzywane jako stosu z danymi.

Definition at line 39 of file `stos_tablica.hh`.

4.2.4.2 `int* stos_tablica::danetmp` `[private]`

Pole typu `int`, bedzie uzywane jako stosu z danymi sprawdzajacymi, pomocniczymi.

Definition at line 43 of file `stos_tablica.hh`.

4.2.4.3 `int stos_tablica::heap_size` `[private]`

Pole typu `int`, bedzie uzywane jako pomocnicza wartosc do metody Heap Size.

Definition at line 63 of file `stos_tablica.hh`.

4.2.4.4 `int stos_tablica::lewy` `[private]`

Pole typu `int`, bedzie uzywane jako pomocnicza wartosc do metody Quick Sort.

Definition at line 55 of file `stos_tablica.hh`.

4.2.4.5 `int stos_tablica::prawy` `[private]`

Pole typu `int`, bedzie uzywane jako pomocnicza wartosc do metody Quick Sort.

Definition at line 59 of file `stos_tablica.hh`.

4.2.4.6 `int stos_tablica::rozmiar` `[private]`

Pole typu `int`, bedzie uzywane jako rozmiar tabeli.

Definition at line 47 of file `stos_tablica.hh`.

4.2.4.7 `int stos_tablica::rozmiar_tmp` `[private]`

Pole typu `int`, bedzie uzywane jako pomocnicza wartosc jako rozmiar stosu.

Definition at line 51 of file stos_tablica.hh.

The documentation for this class was generated from the following files:

- /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/inc/stos_tablica.hh
- /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/stos_tablica.cpp

Chapter 5

File Documentation

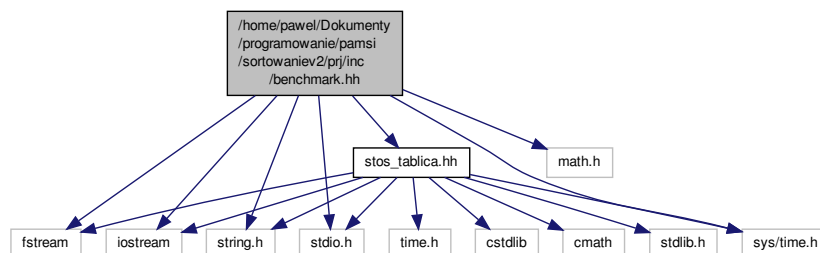
5.1 /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/doc/pages/strona.dox File Reference

5.2 /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/inc/benchmark.hh File Reference

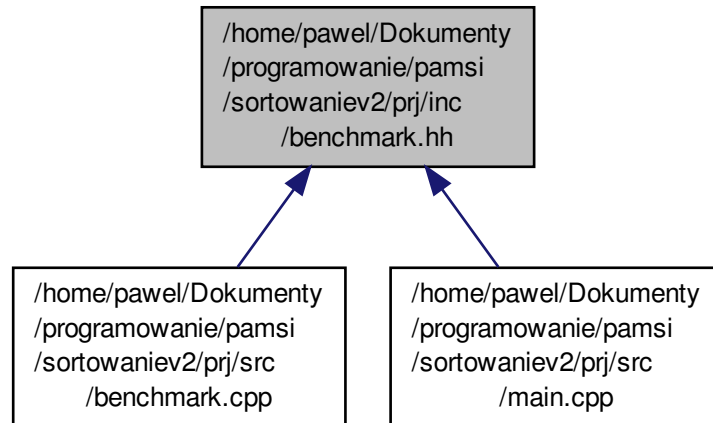
Definicje funkcji dla klasy benchmark.

```
#include <fstream>
#include <iostream>
#include <math.h>
#include <string.h>
#include <stdio.h>
#include <sys/time.h>
#include "stos_tablica.hh"
```

Include dependency graph for benchmark.hh:



This graph shows which files directly or indirectly include this file:



Classes

- class [benchmark](#)

Modeluje pojecie Benchmark.

5.2.1 Detailed Description

Definicje funkcji dla klasy benchmark.

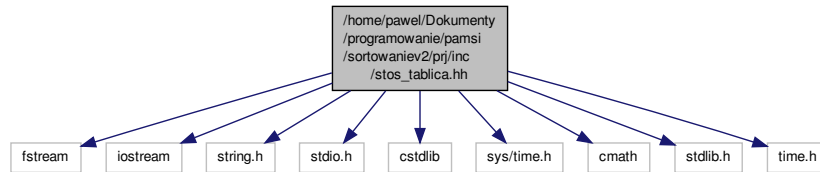
Definition in file [benchmark.hh](#).

5.3 /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/inc/stos_tablica.hh File Reference

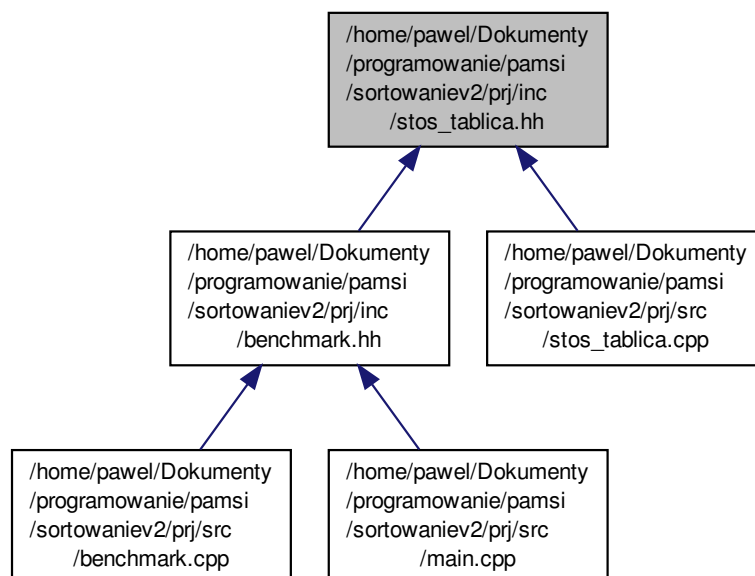
Definicje funkcji dla klasy Stos zdefiniowanej tablica.

```
#include <fstream>
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <cstdlib>
#include <sys/time.h>
#include <cmath>
#include "stdlib.h"
#include "time.h"
```

Include dependency graph for `stos_tablica.hh`:



This graph shows which files directly or indirectly include this file:



Classes

- class [stos_tablica](#)

Modeluje pojecie Stos.

5.3.1 Detailed Description

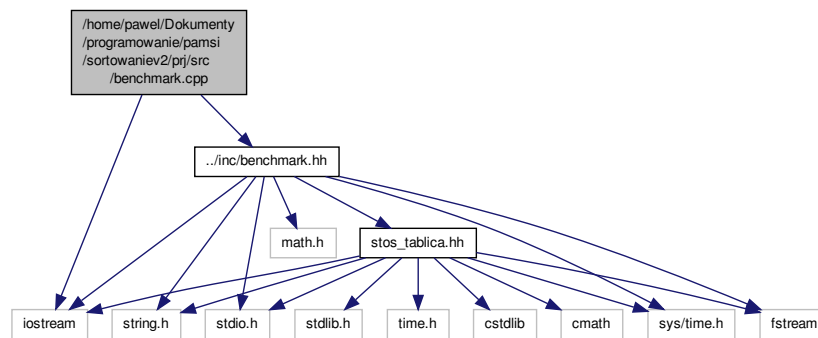
Definicje funkcji dla klasy `Stos` zdefiniowanej tablica.

Definition in file [stos_tablica.hh](#).

5.4 /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/benchmark.cpp File Reference

Plik zawiera funkcje z klasy benchmark.

```
#include "../inc/benchmark.hh"
#include <iostream>
Include dependency graph for benchmark.cpp:
```



5.4.1 Detailed Description

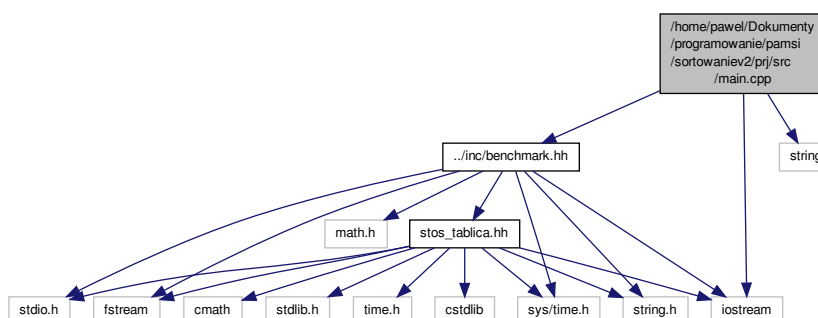
Plik zawiera funkcje z klasy benchmark.

Definition in file [benchmark.cpp](#).

5.5 /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/main.cpp File Reference

Plik zawiera funkcje [main\(\)](#)

```
#include <iostream>
#include <string>
#include "../inc/benchmark.hh"
Include dependency graph for main.cpp:
```



Functions

- int [main](#) ()

5.5.1 Detailed Description

Plik zawiera funkcje [main\(\)](#)

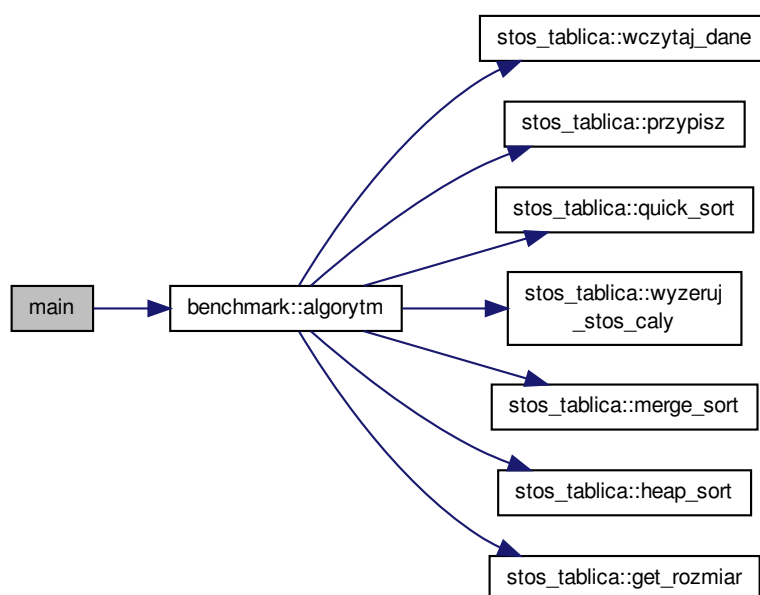
Definition in file [main.cpp](#).

5.5.2 Function Documentation

5.5.2.1 int main ()

Definition at line 13 of file main.cpp.

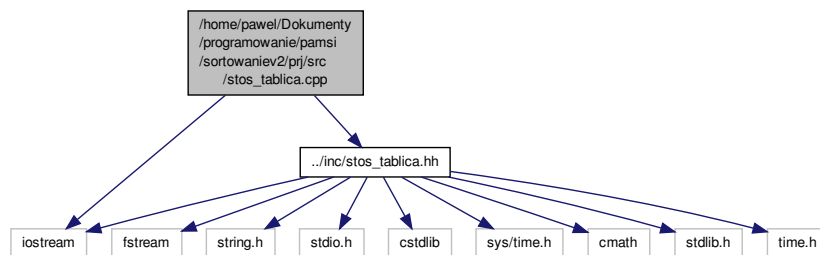
Here is the call graph for this function:



5.6 /home/pawel/Dokumenty/programowanie/pamsi/sortowaniev2/prj/src/stos_tablica.cpp File Reference

```
#include "../inc/stos_tablica.hh"  
#include <iostream>
```

Include dependency graph for stos_tablica.cpp:



Index

~stos_tablica
 stos_tablica, 14
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie2/prj/doc/pages/strona.-
 dox, 21
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie2/prj/doc/benchmark.-
 hh, 21
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie2/prj/src/stos-
 _tablica.hh, 22
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie2/prj/src/benchmark.-
 cpp, 24
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie2/prj/src/main.-
 cpp, 24
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie2/prj/src/stos-
 _tablica.cpp, 25
benchmark, 12
 get_low, 12
 get_prawy, 14
 get_rozmiar, 14
 stos_tablica, 14
 stos_sort, 19
 stos_tablica, 15
 ile_czasu
 benchmark, 10
 lewy
 stos_tablica, 19
 main
 main.cpp, 25
 main.cpp
 main, 25
 merge_sort
 stos_tablica, 15
 mergesort
 stos_tablica, 16
 pokaz_elementy
 stos_tablica, 16
 prawy
 stos_tablica, 19
 przypisz
 stos_tablica, 16
 quick_sort
 stos_tablica, 16
 quicksort
 stos_tablica, 17
 rozmiar
 stos_tablica, 19
 rozmiar_tmp
 stos_tablica, 19
 stos_tablica, 12
 ~stos_tablica, 14
 build_heap, 14
 dane, 19
 danetmp, 19
 disassemble_heap
 stos_tablica, 14
 elapsedTime
 benchmark, 12
 stos_tablica, 14
 stos_sort, 19
 stos_tablica, 15
 ile_czasu
 benchmark, 10
 lewy
 stos_tablica, 19
 main
 main.cpp, 25
 main.cpp
 main, 25
 merge_sort
 stos_tablica, 15
 mergesort
 stos_tablica, 16
 pokaz_elementy
 stos_tablica, 16
 prawy
 stos_tablica, 19
 przypisz
 stos_tablica, 16
 quick_sort
 stos_tablica, 16
 quicksort
 stos_tablica, 17
 rozmiar
 stos_tablica, 19
 rozmiar_tmp
 stos_tablica, 19
 stos_tablica, 12
 ~stos_tablica, 14
 build_heap, 14
 dane, 19
 danetmp, 19
algoritm
 benchmark, 8
benchmark, 7
 algoritm, 8
 benchmark, 8
 czas, 11
 czas_caly, 11
 czas_start, 9
 czas_stop, 9
 elapsedTime, 12
 ile_czasu, 10
 t1, 12
 t2, 12
 wykonaj_algoritm_sortowanie, 10
 wyswietl_wszystko, 11
build_heap
 stos_tablica, 14
czas
 benchmark, 11
czas_caly
 benchmark, 11
czas_start
 benchmark, 9
czas_stop
 benchmark, 9
dane
 stos_tablica, 19
danetmp
 stos_tablica, 19
disassemble_heap
 stos_tablica, 14
elapsedTime

- disassemble_heap, [14](#)
- get_lewy, [14](#)
- get_prawy, [14](#)
- get_rozmiar, [14](#)
- heap_size, [19](#)
- heap_sort, [15](#)
- lewy, [19](#)
- merge_sort, [15](#)
- mergesort, [16](#)
- pokaz_elementy, [16](#)
- prawy, [19](#)
- przypisz, [16](#)
- quick_sort, [16](#)
- quicksort, [17](#)
- rozmiar, [19](#)
- rozmiar_tmp, [19](#)
- stos_tablica, [14](#)
- stos_tablica, [14](#)
- swap, [17](#)
- wczytaj_dane, [17](#), [18](#)
- wypelnij_losowo, [18](#)
- wyzeruj_stos, [18](#)
- wyzeruj_stos_caly, [18](#)
- swap
 - stos_tablica, [17](#)
- t1
 - benchmark, [12](#)
- t2
 - benchmark, [12](#)
- wczytaj_dane
 - stos_tablica, [17](#), [18](#)
- wykonaj_algorytm_sortowanie
 - benchmark, [10](#)
- wypelnij_losowo
 - stos_tablica, [18](#)
- wyswietl_wszystko
 - benchmark, [11](#)
- wyzeruj_stos
 - stos_tablica, [18](#)
- wyzeruj_stos_caly
 - stos_tablica, [18](#)