

# Sortowanie

1.0

Generated by Doxygen 1.8.4

Tue Mar 25 2014 19:51:37



# Contents

<b>1</b>	<b>Sortowanie</b>	<b>1</b>
1.1	Najważniejsze cechy	1
1.2	Opis algorytmu	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	benchmark Class Reference	7
4.1.1	Detailed Description	7
4.1.2	Constructor & Destructor Documentation	8
4.1.2.1	benchmark	8
4.1.3	Member Function Documentation	8
4.1.3.1	czas_start	8
4.1.3.2	czas_stop	8
4.1.3.3	ile_czasu	8
4.1.3.4	wykonaj_algorytm_sortowanie	9
4.1.4	Member Data Documentation	10
4.1.4.1	czas	10
4.1.4.2	elapsedTime	10
4.1.4.3	t1	10
4.1.4.4	t2	10
4.2	stos_tablica Class Reference	10
4.2.1	Detailed Description	12
4.2.2	Constructor & Destructor Documentation	12
4.2.2.1	stos_tablica	12
4.2.2.2	~stos_tablica	12
4.2.3	Member Function Documentation	12
4.2.3.1	build_heap	12

4.2.3.2	<a href="#">disassemble_heap</a>	12
4.2.3.3	<a href="#">get_lewy</a>	12
4.2.3.4	<a href="#">get_prawy</a>	12
4.2.3.5	<a href="#">get_rozmiar</a>	13
4.2.3.6	<a href="#">heap_sort</a>	13
4.2.3.7	<a href="#">merge_sort</a>	13
4.2.3.8	<a href="#">mergesort</a>	14
4.2.3.9	<a href="#">pokaz_elementy</a>	14
4.2.3.10	<a href="#">przypisz</a>	14
4.2.3.11	<a href="#">quick_sort</a>	14
4.2.3.12	<a href="#">quicksort</a>	15
4.2.3.13	<a href="#">swap</a>	15
4.2.3.14	<a href="#">wczytaj_dane</a>	15
4.2.3.15	<a href="#">wczytaj_dane</a>	15
4.2.3.16	<a href="#">wypelnij_losowo</a>	16
4.2.3.17	<a href="#">wyzeruj_stos</a>	16
4.2.3.18	<a href="#">wyzeruj_stos_caly</a>	16
4.2.4	<a href="#">Member Data Documentation</a>	16
4.2.4.1	<a href="#">dane</a>	16
4.2.4.2	<a href="#">danetmp</a>	16
4.2.4.3	<a href="#">heap_size</a>	16
4.2.4.4	<a href="#">lewy</a>	17
4.2.4.5	<a href="#">prawy</a>	17
4.2.4.6	<a href="#">rozmiar</a>	17
4.2.4.7	<a href="#">rozmiar_tmp</a>	17
<b>5</b>	<b><a href="#">File Documentation</a></b>	<b>19</b>
5.1	<a href="#">/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/doc/pages/strona.dox File Reference</a>	19
5.2	<a href="#">/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/benchmark.hh File Reference</a>	19
5.2.1	<a href="#">Detailed Description</a>	20
5.3	<a href="#">/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/stos_tablica.hh File Reference</a>	20
5.3.1	<a href="#">Detailed Description</a>	21
5.4	<a href="#">/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/benchmark.cpp File Reference</a>	21
5.4.1	<a href="#">Detailed Description</a>	22
5.5	<a href="#">/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/main.cpp File Reference</a>	22
5.5.1	<a href="#">Detailed Description</a>	23
5.5.2	<a href="#">Function Documentation</a>	23
5.5.2.1	<a href="#">main</a>	23
5.6	<a href="#">/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/stos_tablica.cpp File Reference</a>	23

**[Index](#)****25**



# Chapter 1

## Sortowanie

### Author

Pawel Zurek

### Date

22.03.2014

### Version

1.0

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 3 petle :

- > 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

Czasy każdego sortowania dla każdej ilości elementów razem z czasem całkowitym są wyprowadzane na standardowe wyjście. Wykonania programu na moim komputerze (różnice w czasach wykonywania sortowań mogą wystąpić, będą najprawdopodobniej spowodowane różnicą w szybkości procesora):



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">benchmark</a>	Modeluje pojecie Benchmark . . . . .	7
<a href="#">stos_tablica</a>	Modeluje pojecie Stos . . . . .	10



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/ <a href="#">benchmark.hh</a>	
Definicje funkcji dla klasy benchmark . . . . .	19
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/ <a href="#">stos_tablica.hh</a>	
Definicje funkcji dla klasy Stos zdefiniowanej tablica . . . . .	20
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/ <a href="#">benchmark.cpp</a>	
Plik zawiera funkcje z klasy benchmark . . . . .	21
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/ <a href="#">main.cpp</a>	
Plik zawiera funkcje <a href="#">main()</a> . . . . .	22
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/ <a href="#">stos_tablica.cpp</a>	23



## 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.*

#### 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.*
- `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 6 zestawu danych o ilosci elementow :

-> 10

-> 100

-> 1000

-> 10000

-> 100000

-> 1000000

Trzema roznymi metodami :

-> Quick Sort

-> Merge Sort

-> Heap Sort

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

Definition at line 40 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 66 of file benchmark.hh.

## 4.1.3 Member Function Documentation

### 4.1.3.1 `void benchmark::czas_start( )`

Funkcja pomocnicza mierzenia czasu.

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

Definition at line 14 of file benchmark.cpp.

### 4.1.3.2 `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.

### 4.1.3.3 `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.

## 4.1.3.4 void benchmark::wykonaj\_algorytm\_sortowanie ( )

Funkcja wykonująca algorytm.

Wykonanie algorytmu ma przebieg :

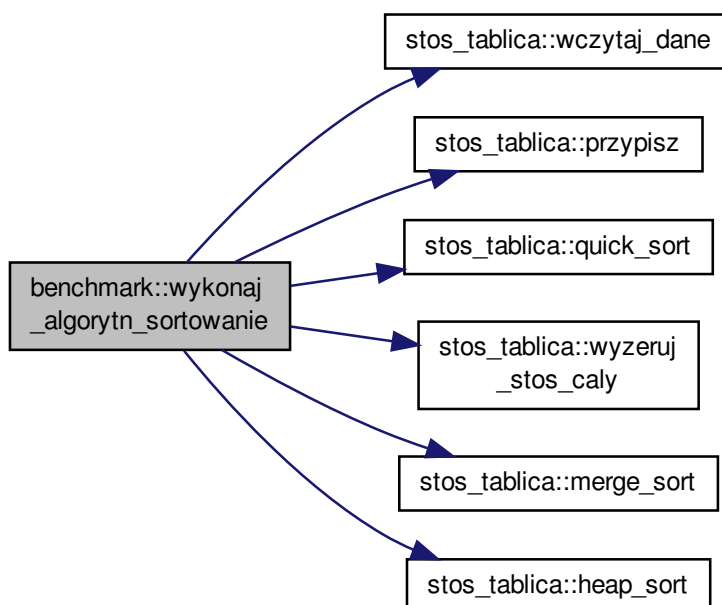
- > posortowanie danych metoda Quick Sort
- > posortowanie danych metoda Merge Sort
- > posortowanie danych metoda Heap Sort

Dla :

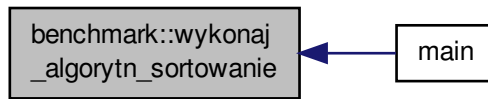
- 10 elementow
- 100 elementow
- 1000 elementow
- 10000 elementow
- 100000 elementow
- 1000000 elementow

Definition at line 27 of file benchmark.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 4.1.4 Member Data Documentation

##### 4.1.4.1 `double benchmark::czas` [private]

Pole typu double, będzie używane do mierzenia całkowitego czasu działania programu.

Definition at line 50 of file benchmark.hh.

##### 4.1.4.2 `double benchmark::elapsedTime` [private]

Pole typu double, będzie używane do mierzenia czasu działania pojedynczego wypełnienia.

Definition at line 46 of file benchmark.hh.

##### 4.1.4.3 `timeval benchmark::t1` [private]

Pole typu timeval, pomoc do liczenia czasu działania operacji krótkich ( tzn pojedynczego działania)

Definition at line 54 of file benchmark.hh.

##### 4.1.4.4 `timeval benchmark::t2` [private]

Definition at line 54 of file benchmark.hh.

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

- [/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/benchmark.hh](#)
- [/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/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 33 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 skladowe klasy wartosciami zerowymi.

Definition at line 73 of file stos\_tablica.hh.

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

Destruktor klasy stos.

Usuwa dynamicznie zaalokowana tablice

Definition at line 80 of file stos\_tablica.hh.

### 4.2.3 Member Function Documentation

#### 4.2.3.1 `void stos_tablica::build_heap ( )`

Funkcja budujaca kopiec ( Heap ).

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

Definition at line 184 of file stos\_tablica.cpp.

#### 4.2.3.2 `void stos_tablica::disassemble_heap ( )`

Funkcja rozbierajaca kopiec ( Heap ).

Funkcja rozbiera kopiec, jednoczesnie sortujac dane.

Definition at line 203 of file stos\_tablica.cpp.

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

Funkcja dostepu do prywatnego pola klasy lewy.

Returns

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

Definition at line 185 of file stos\_tablica.hh.

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

Funkcja dostepu do prywatnego pola klasy prawy.

**Returns**

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

Definition at line 192 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 127 of file stos\_tablica.hh.

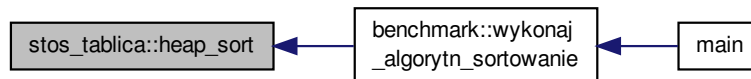
**4.2.3.6 void stos\_tablica::heap\_sort ( ) [inline]**

Funkcja wywołująca funkcję sortującą Heap Sort.

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

Definition at line 233 of file stos\_tablica.hh.

Here is the caller graph for this function:

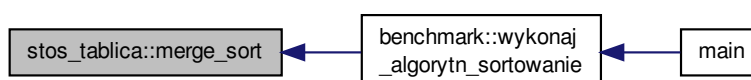
**4.2.3.7 void stos\_tablica::merge\_sort ( ) [inline]**

Funkcja wywołująca funkcję sortującą Merge Sort.

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

Definition at line 210 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 160 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.

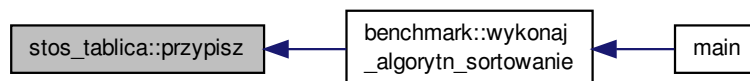
#### 4.2.3.10 void stos\_tablica::przypisz ( )

Funkcja przypisujaca.

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

Definition at line 64 of file stos\_tablica.cpp.

Here is the caller graph for this function:



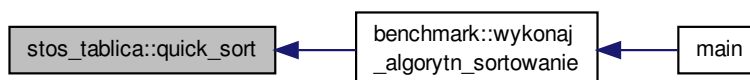
#### 4.2.3.11 void stos\_tablica::quick\_sort ( ) [inline]

Funkcja wywolujaca funkcje sorujaca Quick Sort.

Funkcja jest tylko po to, aby mozna bylo w kklasie innej niz ta uruchomic funkcje sortowania bezparametrycznie

Definition at line 145 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 sortujaca.

Funkcja sortuje dane za pomoca algorytmu Quick Sort

**Parameters**

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

Definition at line 126 of file stos\_tablica.cpp.

**4.2.3.13 void stos\_tablica::swap ( int \* *x*, int \* *y* )**

Funkcja zamieniania wartosci.

Funkcja zamienia wartosciami obiekty, ktore sa argumentami funkcji

**Parameters**

<i>x</i>	-> Pole typu *int, zawiera adres obiektu, ktorego wartosc ma byc zamieniona z obiektem <i>y</i>
<i>y</i>	-> Pole typu *int, zawiera adres obiektu, ktorego wartosc ma byc zamieniona z obiektem <i>x</i>

Definition at line 153 of file stos\_tablica.cpp.

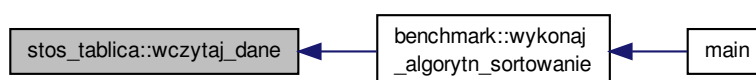
**4.2.3.14 void stos\_tablica::wczytaj\_dane ( )**

Funkcja wczytywania.

Funkcja wczytuje wartosci do tabeli z podanego pliku przez uzytkownika.

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 79 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 99 of file stos\_tablica.cpp.

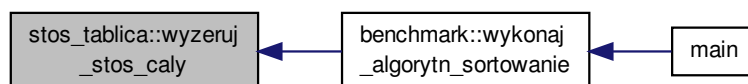
#### 4.2.3.18 void stos\_tablica::wyzeruj\_stos\_caly ( )

Funkcja zerujaca caly stos.

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

Definition at line 110 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 38 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 42 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 62 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 54 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 58 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 46 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 50 of file stos\_tablica.hh.

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

- [/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/stos\\_tablica.hh](/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/stos_tablica.hh)
- [/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/stos\\_tablica.cpp](/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/stos_tablica.cpp)





## Chapter 5

# File Documentation

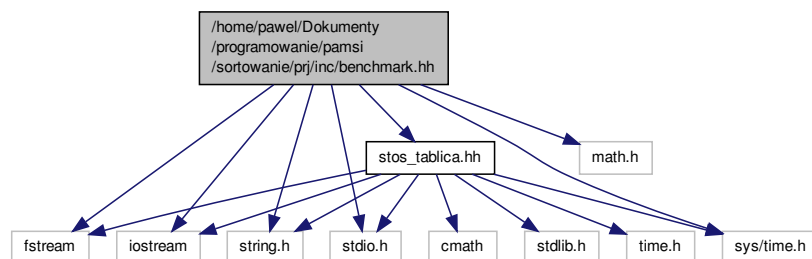
### 5.1 /home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/doc/pages/strona.dox File Reference

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

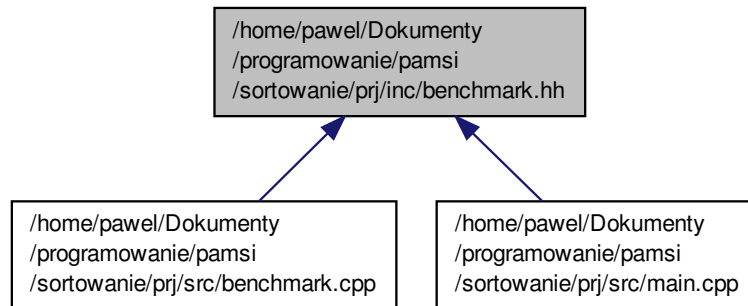
Definicje funkcji dla klasy benchmark.

```
#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`.

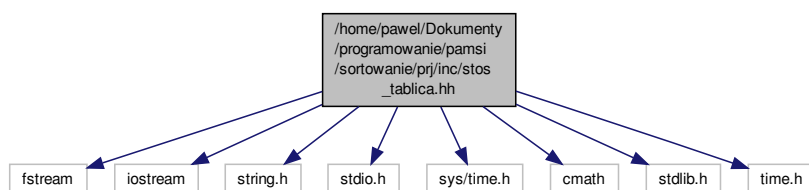
Definition in file [benchmark.hh](#).

## 5.3 `/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/stos_tablica.hh` File Reference

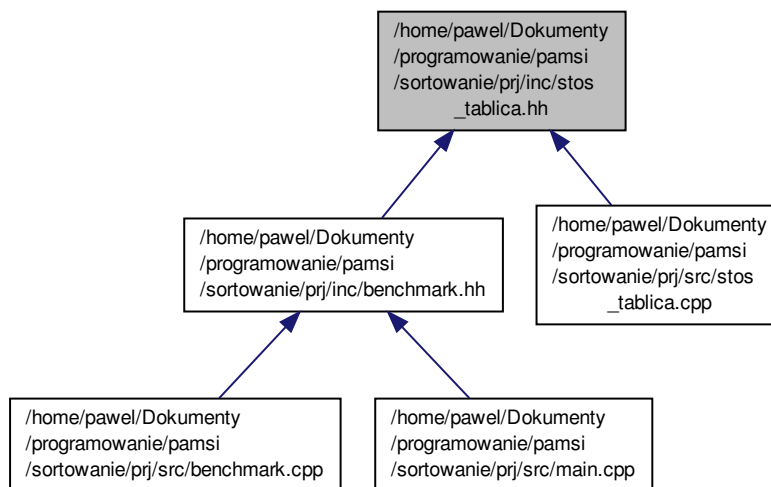
Definicje funkcji dla klasy `Stos` zdefiniowanej tablica.

```
#include <fstream>
#include <iostream>
#include <string.h>
#include <stdio.h>
#include <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.

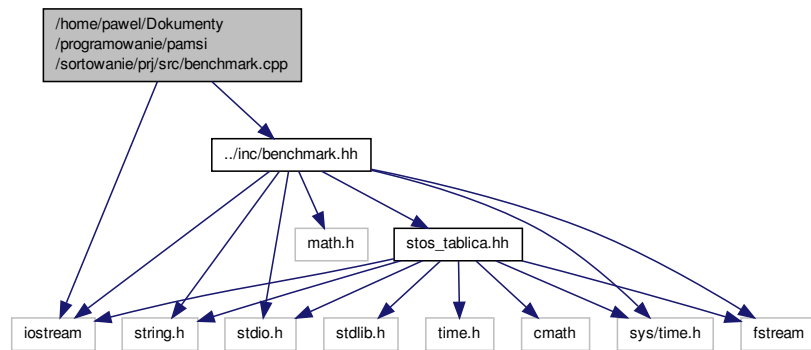
Definition in file [stos\\_tablica.hh](#).

## 5.4 /home/pawel/Dokumenty/programowanie/pamsi/sortowanie/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.

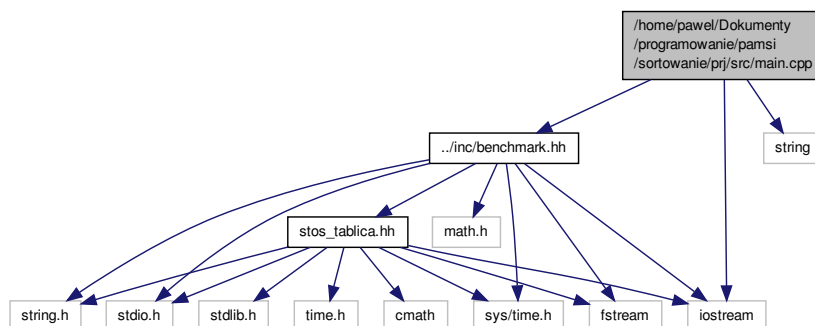
Definition in file [benchmark.cpp](#).

## 5.5 /home/pawel/Dokumenty/programowanie/pamsi/sortowanie/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()`

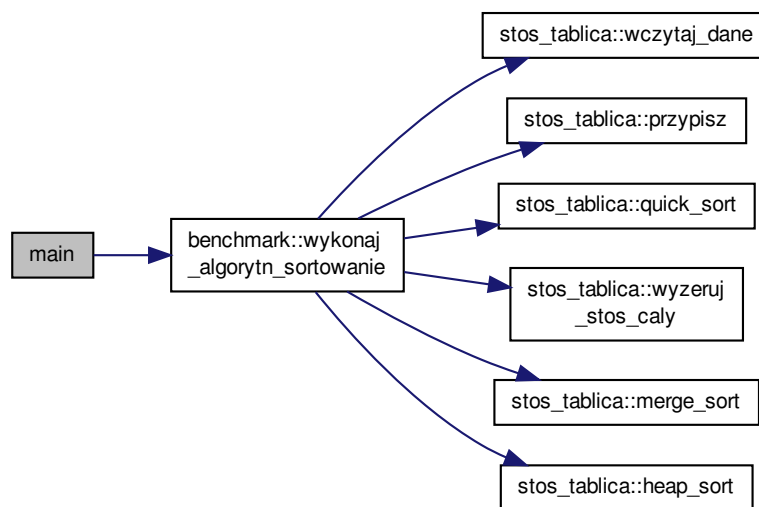
Definition in file `main.cpp`.

### 5.5.2 Function Documentation

#### 5.5.2.1 `int main ( )`

Definition at line 13 of file `main.cpp`.

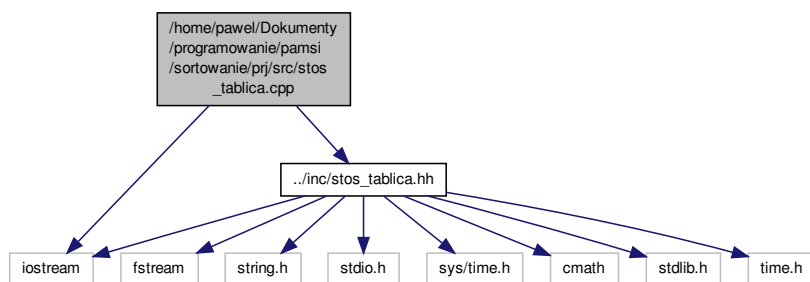
Here is the call graph for this function:



## 5.6 /home/pawel/Dokumenty/programowanie/pamsi/sortowanie/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, 12  
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/pages/strona.-  
  dox, 19  
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/benchmark.-  
  hh, 19  
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/inc/stos-  
  \_tablica.hh, 20  
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/benchmark.-  
  cpp, 21  
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/main.-  
  cpp, 22  
/home/pawel/Dokumenty/programowanie/pamsi/sortowanie/prj/src/stos-  
  \_tablica.cpp, 23  
benchmark, 7  
  benchmark, 8  
  czas, 10  
  czas\_start, 8  
  czas\_stop, 8  
  elapsedTime, 10  
  ile\_czasu, 8  
  t1, 10  
  t2, 10  
  wykonaj\_algorytm\_sortowanie, 8  
build\_heap  
  stos\_tablica, 12  
czas  
  benchmark, 10  
czas\_start  
  benchmark, 8  
czas\_stop  
  benchmark, 8  
dane  
  stos\_tablica, 16  
danetmp  
  stos\_tablica, 16  
disassemble\_heap  
  stos\_tablica, 12  
elapsedTime  
  benchmark, 10  
get\_lewy  
  stos\_tablica, 12  
get\_prawy  
  stos\_tablica, 12  
get\_rozmiar  
  stos\_tablica, 13  
heap\_size  
  stos\_tablica, 16  
heap\_sort  
  stos\_tablica, 13  
ile\_czasu  
  benchmark, 8  
lewy  
  stos\_tablica, 17  
main  
  main.cpp, 23  
  main.cpp  
  main, 23  
merge\_sort  
  stos\_tablica, 13  
mergesort  
  stos\_tablica, 13  
pokaz\_elementy  
  stos\_tablica, 14  
prawy  
  stos\_tablica, 17  
przypisz  
  stos\_tablica, 14  
quick\_sort  
  stos\_tablica, 14  
quicksort  
  stos\_tablica, 15  
rozmiar  
  stos\_tablica, 17  
rozmiar\_tmp  
  stos\_tablica, 17  
stos\_tablica, 10  
  ~stos\_tablica, 12  
  build\_heap, 12  
  dane, 16  
  danetmp, 16  
  disassemble\_heap, 12  
  get\_lewy, 12  
  get\_prawy, 12  
  get\_rozmiar, 13  
  heap\_size, 16  
  heap\_sort, 13  
  lewy, 17  
  merge\_sort, 13  
  mergesort, 13

- [pokaz\\_elementy](#), [14](#)
  - [prawy](#), [17](#)
  - [przypisz](#), [14](#)
  - [quick\\_sort](#), [14](#)
  - [quicksort](#), [15](#)
  - [rozmiar](#), [17](#)
  - [rozmiar\\_tmp](#), [17](#)
  - [stos\\_tablica](#), [12](#)
  - [stos\\_tablica](#), [12](#)
  - [swap](#), [15](#)
  - [wczytaj\\_dane](#), [15](#)
  - [wypelnij\\_losowo](#), [16](#)
  - [wyzeruj\\_stos](#), [16](#)
  - [wyzeruj\\_stos\\_caly](#), [16](#)
- [swap](#)
  - [stos\\_tablica](#), [15](#)
- [t1](#)
  - [benchmark](#), [10](#)
- [t2](#)
  - [benchmark](#), [10](#)
- [wczytaj\\_dane](#)
  - [stos\\_tablica](#), [15](#)
- [wykonaj\\_algorytm\\_sortowanie](#)
  - [benchmark](#), [8](#)
- [wypelnij\\_losowo](#)
  - [stos\\_tablica](#), [16](#)
- [wyzeruj\\_stos](#)
  - [stos\\_tablica](#), [16](#)
- [wyzeruj\\_stos\\_caly](#)
  - [stos\\_tablica](#), [16](#)