MergeSort

Wygenerowano za pomocą Doxygen 1.12.0

1 Lista testów 1

2 Indeks klas 3

2.1 Lista klas . . . 3

3 Indeks plików 5

3.1 Lista plików . . . 5

4 Dokumentacja klas 7

4.1 Dokumentacja klasy MergeSort . . . 7

4.1.1 Opis szczegółowy . . . 7

4.1.2 Dokumentacja funkcji składowych . . . 7

4.1.2.1 merge() . . . 7

4.1.2.2 mergeSort() . . . 8

4.1.2.3 sort() . . . 8

5 Dokumentacja plików 11

5.1 Dokumentacja pliku MergeSort.cpp . . . 11

5.1.1 Dokumentacja funkcji . . . 11

5.1.1.1 main() . . . 11

5.2 Dokumentacja pliku MergeSortclass.cpp . . . 12

5.3 Dokumentacja pliku MergeStoreclass.hpp . . . 12

5.4 MergeStoreclass.hpp . . . 12

5.5 Dokumentacja pliku testy/testdoduje.cpp . . . 12

5.5.1 Dokumentacja funkcji . . . 13

5.5.1.1 TEST() . . . 13

5.6 Dokumentacja pliku testy/testduplikat.cpp . . . 13

5.6.1 Dokumentacja funkcji . . . 13

5.6.1.1 TEST() . . . 13

5.7 Dokumentacja pliku testy/testjeden.cpp . . . 14

5.7.1 Dokumentacja funkcji . . . 14

5.7.1.1 TEST() . . . 14

5.8 Dokumentacja pliku testy/testlosowy.cpp . . . 14

5.8.1 Dokumentacja funkcji . . . 15

5.8.1.1 TEST() . . . 15

5.9 Dokumentacja pliku testy/testodwrotny.cpp . . . 15

5.9.1 Dokumentacja funkcji . . . 15

5.9.1.1 TEST() . . . 15

5.10 Dokumentacja pliku testy/testpusty.cpp . . . 16

5.10.1 Dokumentacja funkcji . . . 16

5.10.1.1 TEST() . . . 16

5.11 Dokumentacja pliku testy/Testsortowania.cpp . . . 16

5.11.1 Dokumentacja funkcji . . . 17

5.11.1.1 TEST() . . . 17

5.12 Dokumentacja pliku testy/testujemny.cpp 17
5.12.1 Dokumentacja funkcji 17
5.12.1.1 TEST() 17
5.13 Dokumentacja pliku testy/testwiekszy.cpp 18
5.13.1 Dokumentacja funkcji 18
5.13.1.1 TEST() 18

Skorowidz **19**

Rozdział 1

Lista testów

Składowa TEST (MergeSortTests, NegativeAndPositiveNumbers)

MergeSortTests.NegativeAndPositiveNumbers

Składowa TEST (MergeSortTests, DuplicatesArray)

MergeSortTests.DuplicatesArray

Składowa TEST (MergeSortTests, SingleElementArray)

MergeSortTests.SingleElementArray

Składowa TEST (MergeSortTests, RandomArray)

MergeSortTests.RandomArray

Składowa TEST (MergeSortTests, ReversedArray)

MergeSortTests.ReversedArray

Składowa TEST (MergeSortTests, EmptyArray)

MergeSortTests.EmptyArray

Składowa TEST (MergeSortTests, SortedArray)

MergeSortTests.SortedArray

Składowa TEST (MergeSortTests, NegativeNumbers)

MergeSortTests.NegativeNumbers

Składowa TEST (MergeSortTests, LargeArray)

MergeSortTests.LargeArray

Rozdział 2

Indeks klas

2.1 Lista klas

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

MergeSort
Klasa implementująca algorytm sortowania przez scalanie (MergeSort) 7

Rozdział 3

Indeks plików

3.1 Lista plików

Tutaj znajduje się lista wszystkich plików wraz z ich krótkimi opisami:

MergeSort.cpp 11
MergeSortclass.cpp 12
MergeStoreclass.hpp 12
testy/testdoduje.cpp 12
testy/testduplikat.cpp 13
testy/testjeden.cpp 14
testy/testlosowy.cpp 14
testy/testodwrotny.cpp 15
testy/testpusty.cpp 16
testy/Testsortowania.cpp 16
testy/testujemny.cpp 17
testy/testwiekszy.cpp 18

Rozdział 4

Dokumentacja klas

4.1 Dokumentacja klasy MergeSort

Klasa implementująca algorytm sortowania przez scalanie (MergeSort).

`#include <MergeStoreclass.hpp>`

Statyczne metody publiczne

- static void sort (std::vector< int > &arr)
  Sortuje przekazaną tablicę za pomocą algorytmu MergeSort.

Statyczne metody prywatne

- static void merge (std::vector< int > &arr, int left, int right)
  Pomocnicza funkcja do scalenia dwóch posortowanych części tablicy.
- static void mergeSort (std::vector< int > &arr, int left, int right)
  Rekursywnie dzieli tablicę na części i sortuje je.

4.1.1 Opis szczegółowy

Klasa implementująca algorytm sortowania przez scalanie (MergeSort).

Klasa ta zawiera metody do sortowania tablicy liczb całkowitych przy użyciu algorytmu MergeSort. Sortowanie jest wykonywane w sposób rekurencyjny, dzieląc tablicę na części, a następnie scalając je w sposób posortowany.

4.1.2 Dokumentacja funkcji składowych

4.1.2.1 merge()

```
void MergeSort::merge (
            std::vector< int > & arr,
            int left,
            int right)  [static], [private]
```

Pomocnicza funkcja do scalenia dwóch posortowanych części tablicy.

Łączy dwie posortowane części tablicy w jedną uporządkowaną część.

Funkcja ta łączy dwie posortowane części tablicy w jedną uporządkowaną część. Jest wywoływana po podziale tablicy na mniejsze części.

Parametry

arr	Tablica, która zawiera części do scalania.
left	Indeks początkowy lewej części tablicy.
right	Indeks końcowy prawej części tablicy.

Funkcja odpowiedzialna za scalanie dwóch części tablicy w jedną, posortowaną część. Działa na przekazanej tablicy, wykorzystując indeksy do określenia zakresów.

Parametry

arr	Tablica liczb całkowitych do scalania.
left	Indeks początkowy zakresu do scalania.
right	Indeks końcowy zakresu do scalania.

4.1.2.2 mergeSort()

```
void MergeSort::mergeSort (
            std::vector< int > & arr,
            int left,
            int right)  [static], [private]
```

Rekursywnie dzieli tablicę na części i sortuje je.

Rekurencyjnie dzieli tablicę na części i sortuje je.

Funkcja ta dzieli tablicę na mniejsze części, aż do momentu, gdy każda część będzie zawierać tylko jeden element. Następnie wywołuje funkcję merge, aby połączyć posortowane części.

Parametry

arr	Tablica do posortowania.
left	Indeks początkowy aktualnej części tablicy.
right	Indeks końcowy aktualnej części tablicy.

Funkcja dzieli tablicę na coraz mniejsze części (podtablice), aż każda będzie zawierała pojedynczy element. Następnie scala je, tworząc uporządkowaną tablicę.

Parametry

arr	Tablica liczb całkowitych do posortowania.
left	Indeks początkowy aktualnego zakresu.
right	Indeks końcowy aktualnego zakresu.

4.1.2.3 sort()

```
void MergeSort::sort (
            std::vector< int > & arr)  [static]
```

Sortuje przekazaną tablicę za pomocą algorytmu MergeSort.

Sortuje całą tablicę liczb całkowitych.

Ta metoda jest wywoływana w celu posortowania całej tablicy. Algorytm działa w sposób rekurencyjny, dzieląc tablicę na mniejsze części i scalając je w sposób uporządkowany.

Parametry

arr	Tablica liczb całkowitych do posortowania.

Zwraca

Brak (sortowanie odbywa się w miejscu na oryginalnej tablicy).

Funkcja jest punktem wejścia do sortowania całej tablicy. Wywołuje rekurencyjną metodę `mergeSort`, która dzieli tablicę na mniejsze części i je sortuje.

Parametry

arr	Tablica liczb całkowitych do posortowania.

Dokumentacja dla tej klasy została wygenerowana z plików:

- MergeStoreclass.hpp
- MergeSortclass.cpp

Rozdział 5

Dokumentacja plików

5.1 Dokumentacja pliku MergeSort.cpp

```
#include <iostream>
#include <vector>
#include "MergeStoreclass.hpp"
```

Funkcje

- int main ()

  Punkt wejścia programu.

5.1.1 Dokumentacja funkcji

5.1.1.1 main()

```
int main ()
```

Punkt wejścia programu.

Funkcja tworzy przykładową tablicę, wypisuje jej elementy, sortuje je za pomocą klasy MergeSort, a następnie wypisuje wynik.

Zwraca

Zwraca 0 w przypadku powodzenia.

Przykładowa tablica liczb całkowitych do posortowania.

Sortowanie tablicy za pomocą algorytmu MergeSort.

5.2 Dokumentacja pliku MergeSortclass.cpp

```
#include "MergeStoreclass.hpp"
```

5.3 Dokumentacja pliku MergeStoreclass.hpp

```
#include <vector>
```

Komponenty

- class MergeSort

  Klasa implementująca algorytm sortowania przez scalanie (MergeSort).

5.4 MergeStoreclass.hpp

Idź do dokumentacji tego pliku.

```
00001 #include <vector>
00002
00012 class MergeSort {
00013 public:
00024     static void sort(std::vector<int>& arr);
00025
00026 private:
00037     static void merge(std::vector<int>& arr, int left, int right);
00038
00050     static void mergeSort(std::vector<int>& arr, int left, int right);
00051 };
00052
```

5.5 Dokumentacja pliku testy/testdoduje.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, NegativeAndPositiveNumbers)

  Testuje poprawność działania algorytmu MergeSort dla tablicy zawierającej zarówno liczby ujemne, jak i dodatnie.

5.5.1 Dokumentacja funkcji

5.5.1.1 TEST()

```
TEST (
            MergeSortTests ,
            NegativeAndPositiveNumbers )
```

Testuje poprawność działania algorytmu MergeSort dla tablicy zawierającej zarówno liczby ujemne, jak i dodatnie.

Test MergeSortTests.NegativeAndPositiveNumbers

Test weryfikuje, czy algorytm sortowania poprawnie obsługuje mieszane dane (ujemne i dodatnie wartości), sortując je w porządku rosnącym.

- Wejście: Tablica {-1, -5, 3, 2, -4}.
- Oczekiwane wyjście: Tablica {-5, -4, -1, 2, 3}.

5.6 Dokumentacja pliku testy/testduplikat.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, DuplicatesArray)
  Testuje działanie algorytmu MergeSort dla tablicy zawierającej duplikaty elementów.

5.6.1 Dokumentacja funkcji

5.6.1.1 TEST()

```
TEST (
            MergeSortTests ,
            DuplicatesArray )
```

Testuje działanie algorytmu MergeSort dla tablicy zawierającej duplikaty elementów.

Test MergeSortTests.DuplicatesArray

Test weryfikuje, czy algorytm sortowania poprawnie obsługuje tablicę, w której znajdują się powtarzające się wartości, sortując je w porządku rosnącym.

- Wejście: Tablica {3, 3, 1, 2, 2}.
- Oczekiwane wyjście: Tablica {1, 2, 2, 3, 3}.

< Tablica wejściowa zawierająca duplikaty.

< Wywołanie algorytmu MergeSort.

< Sprawdzenie, czy wynik jest zgodny z oczekiwaniami.

5.7 Dokumentacja pliku testy/testjeden.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, SingleElementArray)

  Testuje działanie algorytmu MergeSort na tablicy zawierającej jeden element.

5.7.1 Dokumentacja funkcji

5.7.1.1 TEST()

```
TEST (
            MergeSortTests ,
            SingleElementArray )
```

Testuje działanie algorytmu MergeSort na tablicy zawierającej jeden element.

Test MergeSortTests.SingleElementArray

Test weryfikuje, czy algorytm poprawnie obsługuje przypadek, w którym tablica wejściowa zawiera tylko jeden element. W takim przypadku tablica powinna pozostać bez zmian.

- Wejście: Tablica {42}.
- Oczekiwane wyjście: Tablica {42}.

< Tablica wejściowa zawierająca jeden element.

< Wywołanie algorytmu MergeSort.

< Sprawdzenie, czy wynik jest zgodny z oczekiwaniami.

5.8 Dokumentacja pliku testy/testlosowy.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, RandomArray)

  Testuje działanie algorytmu MergeSort na tablicy o losowej kolejności elementów.

5.8.1 Dokumentacja funkcji

5.8.1.1 TEST()

```
TEST (
            MergeSortTests ,
            RandomArray )
```

Testuje działanie algorytmu MergeSort na tablicy o losowej kolejności elementów.

Test MergeSortTests.RandomArray

Test weryfikuje, czy algorytm poprawnie sortuje tablicę zawierającą elementy w przypadkowej kolejności. Oczekiwanym wynikiem jest posortowana tablica w porządku rosnącym.

- Wejście: Tablica {38, 27, 43, 3, 9, 82, 10}.
- Oczekiwane wyjście: Tablica {3, 9, 10, 27, 38, 43, 82}.

< Tablica wejściowa z losową kolejnością elementów.

< Wywołanie algorytmu MergeSort.

< Weryfikacja wyniku sortowania.

5.9 Dokumentacja pliku testy/testodwrotny.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, ReversedArray)
  Testuje działanie algorytmu MergeSort na tablicy w odwrotnej kolejności.

5.9.1 Dokumentacja funkcji

5.9.1.1 TEST()

```
TEST (
            MergeSortTests ,
            ReversedArray )
```

Testuje działanie algorytmu MergeSort na tablicy w odwrotnej kolejności.

Test MergeSortTests.ReversedArray

Test sprawdza, czy algorytm prawidłowo sortuje tablicę, w której elementy są uporządkowane w porządku malejącym. Po sortowaniu elementy w tablicy powinny być uporządkowane rosnąco.

- Wejście: Tablica {5, 4, 3, 2, 1}.
- Oczekiwane wyjście: Tablica {1, 2, 3, 4, 5}.

< Tablica wejściowa z elementami w odwrotnej kolejności.

< Wywołanie algorytmu MergeSort.

< Sprawdzenie, czy wynik jest zgodny z oczekiwaniami.

5.10 Dokumentacja pliku testy/testpusty.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, EmptyArray)

  Testuje działanie algorytmu MergeSort na pustej tablicy.

5.10.1 Dokumentacja funkcji

5.10.1.1 TEST()

```
TEST (
            MergeSortTests ,
            EmptyArray )
```

Testuje działanie algorytmu MergeSort na pustej tablicy.

Test MergeSortTests.EmptyArray

Test sprawdza, czy algorytm prawidłowo obsługuje przypadek pustej tablicy. W przypadku pustej tablicy algorytm nie powinien wprowadzać żadnych zmian, a wynikowa tablica powinna pozostać pusta.

- Wejście: Pusta tablica {}.
- Oczekiwane wyjście: Pusta tablica {}.

< Pusta tablica wejściowa.

< Wywołanie algorytmu MergeSort.

< Sprawdzenie, czy wynik jest pustą tablicą.

5.11 Dokumentacja pliku testy/Testsortowania.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, SortedArray)

  Testuje działanie algorytmu MergeSort na już posortowanej tablicy.

5.11.1 Dokumentacja funkcji

5.11.1.1 TEST()

```
TEST (
            MergeSortTests ,
            SortedArray )
```

Testuje działanie algorytmu MergeSort na już posortowanej tablicy.

Test MergeSortTests.SortedArray

Test sprawdza, czy algorytm prawidłowo obsługuje przypadek tablicy, która jest już posortowana. W takim przypadku algorytm nie powinien wprowadzać żadnych zmian w tablicy.

- Wejście: Tablica już posortowana {1, 2, 3, 4, 5}.
- Oczekiwane wyjście: Tablica pozostaje bez zmian {1, 2, 3, 4, 5}.

< Już posortowana tablica wejściowa.

< Wywołanie algorytmu MergeSort.

< Sprawdzenie, czy wynik jest taki sam jak wejście.

5.12 Dokumentacja pliku testy/testujemny.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, NegativeNumbers)
  Testuje działanie algorytmu MergeSort na tablicy zawierającej liczby ujemne.

5.12.1 Dokumentacja funkcji

5.12.1.1 TEST()

```
TEST (
            MergeSortTests ,
            NegativeNumbers )
```

Testuje działanie algorytmu MergeSort na tablicy zawierającej liczby ujemne.

Test MergeSortTests.NegativeNumbers

Test sprawdza, czy algorytm prawidłowo sortuje tablicę zawierającą liczby ujemne. Po posortowaniu liczby powinny być uporządkowane rosnąco.

- Wejście: Tablica zawierająca liczby ujemne {-1, -5, -3, -2, -4}.
- Oczekiwane wyjście: Tablica posortowana w porządku rosnącym {-5, -4, -3, -2, -1}.

< Tablica wejściowa z liczbami ujemnymi.

< Wywołanie algorytmu MergeSort.

< Sprawdzenie, czy tablica jest posortowana.

5.13 Dokumentacja pliku testy/testwiekszy.cpp

```
#include <vector>
#include <gtest/gtest.h>
#include "MergeStoreclass.hpp"
```

Funkcje

- TEST (MergeSortTests, LargeArray)

  Testuje działanie algorytmu MergeSort na większej tablicy.

5.13.1 Dokumentacja funkcji

5.13.1.1 TEST()

```
TEST (
            MergeSortTests ,
            LargeArray )
```

Testuje działanie algorytmu MergeSort na większej tablicy.

Test MergeSortTests.LargeArray

Test sprawdza, czy algorytm prawidłowo sortuje większą tablicę z różnymi liczbami. Po posortowaniu elementy powinny być uporządkowane rosnąco.

- Wejście: Tablica {100, 30, 10, 90, 60, 50, 20, 80, 70, 40}.
- Oczekiwane wyjście: Tablica posortowana w porządku rosnącym {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}.

< Tablica wejściowa z większymi liczbami.

< Wywołanie algorytmu MergeSort.

< Sprawdzenie, czy tablica jest posortowana.

Skorowidz

Lista testów, 1

main
    MergeSort.cpp, 11
merge
    MergeSort, 7
MergeSort, 7
    merge, 7
    mergeSort, 8
    sort, 8
mergeSort
    MergeSort, 8
MergeSort.cpp, 11
    main, 11
MergeSortclass.cpp, 12
MergeStoreclass.hpp, 12

sort
    MergeSort, 8

TEST
    testdoduje.cpp, 13
    testduplikat.cpp, 13
    testjeden.cpp, 14
    testlosowy.cpp, 15
    testodwrotny.cpp, 15
    testpusty.cpp, 16
    Testsortowania.cpp, 17
    testujemny.cpp, 17
    testwiekszy.cpp, 18
testdoduje.cpp
    TEST, 13
testduplikat.cpp
    TEST, 13
testjeden.cpp
    TEST, 14
testlosowy.cpp
    TEST, 15
testodwrotny.cpp
    TEST, 15
testpusty.cpp
    TEST, 16
Testsortowania.cpp
    TEST, 17
testujemny.cpp
    TEST, 17
testwiekszy.cpp
    TEST, 18
testy/testdoduje.cpp, 12
testy/testduplikat.cpp, 13
testy/testjeden.cpp, 14
testy/testlosowy.cpp, 14
testy/testodwrotny.cpp, 15
testy/testpusty.cpp, 16
testy/Testsortowania.cpp, 16
testy/testujemny.cpp, 17
testy/testwiekszy.cpp, 18