My Project

Generated by Doxygen 1.15.0

Chapter 1

projekt-MergeSort

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 MergeSorter< T > Class Template Reference

Klasa implementująca algorytm sortowania przez scalanie (Merge Sort).

`#include <MergeSorter.h>`

Public Member Functions

- void sort (std::vector< T > &arr)

  Główna metoda publiczna rozpoczynająca proces sortowania.

5.1.1 Detailed Description

template<typename T>
class MergeSorter< T >

Klasa implementująca algorytm sortowania przez scalanie (Merge Sort).

- Klasa wykorzystuje szablony (templates), dzięki czemu może sortować wektory dowolnego typu danych, który obsługuje operatory porównania (<, <=).

- **Template Parameters**

T	Typ danych przechowywanych w wektorze (np. int, double, float).

5.1.2 Member Function Documentation

5.1.2.1 sort()

```
template<typename T>
void MergeSorter< T >::sort (
            std::vector< T > & arr)  [inline]
```

Główna metoda publiczna rozpoczynająca proces sortowania.

- Metoda sprawdza, czy wektor nie jest pusty, a następnie wywołuje rekurencyjną metodę pomocniczą. Sortowanie odbywa się "w miejscu" (na przekazanej tablicy).

- **Parameters**

arr	Referencja do wektora std::vector<T>, który ma zostać posortowany.

Note

- Złożoność czasowa algorytmu wynosi O(n log n) w każdym przypadku.
  Złożoność pamięciowa wynosi O(n) ze względu na tablice pomocnicze.

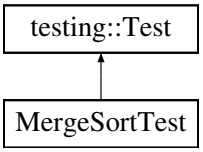
The documentation for this class was generated from the following file:

- MergeSorter.h

5.2 MergeSortTest Class Reference

Fixture testowy (kontekst) dla Google Test.

Inheritance diagram for MergeSortTest:



Protected Attributes

- MergeSorter< int > **sorterInt**
  Instancja do testowania int.
- MergeSorter< double > **sorterDouble**
  Instancja do testowania double.

5.2.1 Detailed Description

Fixture testowy (kontekst) dla Google Test.

- Pozwala zdefiniować obiekty (instancje sorterów), które będą dostępne w każdym teście, co ogranicza powtarzanie kodu.

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

- test.cpp

Chapter 6

File Documentation

6.1 main.cpp File Reference

Główny plik programu demonstrujący działanie algorytmu Merge Sort.

```
#include <iostream>
#include <vector>
#include "MergeSorter.h"
```

Functions

- template<typename T>
  void printVector (const std::vector< T > &vec)

  Szablonowa funkcja pomocnicza do wypisywania zawartości wektora na konsolę.
- int main ()

  Główna funkcja programu.

6.1.1 Detailed Description

Główny plik programu demonstrujący działanie algorytmu Merge Sort.

- Zawiera funkcję main oraz funkcje pomocnicze do wyświetlania wektorów. Demonstruje sortowanie liczb całkowitych (int) i zmiennoprzecinkowych (double).

6.1.2 Function Documentation

6.1.2.1 main()

```
int main ()
```

Główna funkcja programu.

- Tworzy instancje wektorów i klasy sortującej, a następnie przeprowadza testowe sortowanie z wypisaniem wyników przed i po operacji.

  Returns

- int Kod zakończenia programu (0 oznacza sukces).

6.1.2.2 printVector()

```
template<typename T>
void printVector (
            const std::vector< T > & vec)
```

Szablonowa funkcja pomocnicza do wypisywania zawartości wektora na konsolę.

- **Template Parameters**

T	Typ danych w wektorze.

Parameters

vec	Wektor do wyświetlenia.

6.2 MergeSorter.h File Reference

Plik nagłówkowy zawierający szablonową klasę sortowania przez scalanie.

```
#include <vector>
#include <iostream>
```

Classes

- class MergeSorter< T >

  Klasa implementująca algorytm sortowania przez scalanie (Merge Sort).

6.2.1 Detailed Description

Plik nagłówkowy zawierający szablonową klasę sortowania przez scalanie.

Author

Twoje Imie Nazwisko

Date

2023-10-27

6.3 MergeSorter.h

Go to the documentation of this file.

```
00001
00007
00008 #ifndef MERGESORTER_H
00009 #define MERGESORTER_H
00010
00011 #include <vector>
00012 #include <iostream>
00013
00021 template <typename T>
00022 class MergeSorter {
00023 public:
00032     void sort(std::vector<T>& arr) {
00033         if (arr.empty()) return;
00034         mergeSortRecursive(arr, 0, static_cast<int>(arr.size()) - 1);
00035     }
00036
00037 private:
00046     void mergeSortRecursive(std::vector<T>& arr, int left, int right) {
00047         if (left >= right) {
00048             return; // Warunek stopu: 0 lub 1 element
00049         }
00050
00051         int mid = left + (right - left) / 2;
00052
00053         mergeSortRecursive(arr, left, mid);
00054         mergeSortRecursive(arr, mid + 1, right);
00055         merge(arr, left, mid, right);
00056     }
00057
00067     void merge(std::vector<T>& arr, int left, int mid, int right) {
00068         int n1 = mid - left + 1;
00069         int n2 = right - mid;
00070
00071         // Tworzenie tablic tymczasowych
00072         std::vector<T> L(n1);
00073         std::vector<T> R(n2);
00074
00075         // Kopiowanie danych
00076         for (int i = 0; i < n1; i++)
00077             L[i] = arr[left + i];
00078         for (int j = 0; j < n2; j++)
00079             R[j] = arr[mid + 1 + j];
00080
00081         // Scalanie
00082         int i = 0; // Indeks początkowy pierwszej podtablicy
00083         int j = 0; // Indeks początkowy drugiej podtablicy
00084         int k = left; // Indeks początkowy scalonej podtablicy
00085
00086         while (i < n1 && j < n2) {
00087             if (L[i] <= R[j]) {
00088                 arr[k] = L[i];
00089                 i++;
00090             } else {
00091                 arr[k] = R[j];
00092                 j++;
00093             }
00094             k++;
00095         }
00096
00097         // Kopiowanie pozostałych elementów
00098         while (i < n1) {
00099             arr[k] = L[i];
00100             i++;
00101             k++;
00102         }
00103         while (j < n2) {
00104             arr[k] = R[j];
00105             j++;
00106             k++;
00107         }
00108     }
00109 };
00110
00111 #endif // MERGESORTER_H
```

6.4 test.cpp File Reference

Plik zawierający pełny zestaw testów jednostkowych dla algorytmu Merge Sort.

```
#include "gtest/gtest.h"
#include "MergeSorter.h"
#include <vector>
#include <algorithm>
#include <random>
```

Classes

- class MergeSortTest

  Fixture testowy (kontekst) dla Google Test.

Functions

- **TEST_F** (MergeSortTest, AlreadySorted)

  Test 1: Zachowuje tablicę niezmienioną, gdy ona jest już posortowana rosnąco.
- **TEST_F** (MergeSortTest, ReverseSorted)

  Test 2: Potrafi posortować tablicę, która jest posortowana w odwrotnej kolejności.
- **TEST_F** (MergeSortTest, RandomArray)

  Test 3: Poprawnie sortuje losową tablicę liczb.
- **TEST_F** (MergeSortTest, OnlyNegative)

  Test 4: Poprawnie sortuje tablice tylko z liczbami ujemnymi.
- **TEST_F** (MergeSortTest, MixedNegativeAndPositive)

  Test 5: Poprawnie sortuje tablice z liczbami ujemnymi i liczbami dodatnimi.
- **TEST_F** (MergeSortTest, EmptyArray)

  Test 6: Obsługuje pustą tablicę bez rzucania wyjątkiem.
- **TEST_F** (MergeSortTest, SingleElement)

  Test 7: Nie zmienia tablicy, która zawiera tylko jeden element.
- **TEST_F** (MergeSortTest, DuplicatesPositive)

  Test 8: Poprawnie sortuje tablicę z duplikatami liczb (dodatnich).
- **TEST_F** (MergeSortTest, DuplicatesNegative)

  Test 9: Poprawnie sortuje tablice ujemną z duplikatami.
- **TEST_F** (MergeSortTest, MixedDuplicates)

  Test 10: Poprawnie sortuje tablice z liczbami ujemnymi, dodatnimi oraz duplikatami.
- **TEST_F** (MergeSortTest, TwoElementsSorted)

  Test 11: Poprawnie sortuje tablicę zawierającą tylko dwa elementy w kolejności rosnącej.
- **TEST_F** (MergeSortTest, LargeArray)

  Test 12: Poprawnie sortuje dużą tablicę zawierającą ponad 100 elementów.
- **TEST_F** (MergeSortTest, LargeArrayMixedDuplicates)

  Test 13: Poprawnie sortuje dużą tablicę (>100 elementów) z liczbami ujemnymi, dodatnimi oraz duplikatami.

6.4.1 Detailed Description

Plik zawierający pełny zestaw testów jednostkowych dla algorytmu Merge Sort.

- Testy pokrywają przypadki standardowe, brzegowe oraz wydajnościowe przy użyciu biblioteki Google Test.

Index