

Sort Methods

Generated by Doxygen 1.9.3

1 Sorting algorithms for sequences	1
1.0.1 Description	1
1.0.2 Usage	1
1.0.3 Author	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Class Documentation	9
5.1 HeapSort< Key > Class Template Reference	9
5.1.1 Constructor & Destructor Documentation	10
5.1.1.1 HeapSort()	10
5.1.2 Member Function Documentation	10
5.1.2.1 Sort()	10
5.2 Insertion< Key > Class Template Reference	11
5.2.1 Constructor & Destructor Documentation	12
5.2.1.1 Insertion()	12
5.2.2 Member Function Documentation	12
5.2.2.1 Sort()	12
5.3 MergeSort< Key > Class Template Reference	13
5.3.1 Constructor & Destructor Documentation	13
5.3.1.1 MergeSort()	14
5.3.2 Member Function Documentation	14
5.3.2.1 Sort()	14
5.4 RadixSort< Key > Class Template Reference	15
5.4.1 Constructor & Destructor Documentation	15
5.4.1.1 RadixSort()	16
5.4.2 Member Function Documentation	16
5.4.2.1 Sort()	16
5.5 ShellSort< Key > Class Template Reference	17
5.5.1 Constructor & Destructor Documentation	17
5.5.1.1 ShellSort()	18
5.5.2 Member Function Documentation	18
5.5.2.1 Sort()	18
5.6 SortMethod< Key > Class Template Reference	19
5.6.1 Constructor & Destructor Documentation	19
5.6.1.1 SortMethod() [1/2]	19
5.6.1.2 SortMethod() [2/2]	19

5.6.2 Member Function Documentation	20
5.6.2.1 Sort()	20
6 File Documentation	21
6.1 include/HeapSort.h File Reference	21
6.1.1 Detailed Description	22
6.2 HeapSort.h	22
6.3 include/Insertion.h File Reference	23
6.3.1 Detailed Description	23
6.4 Insertion.h	24
6.5 include/MergeSort.h File Reference	24
6.5.1 Detailed Description	25
6.6 MergeSort.h	26
6.7 include/RadixSort.h File Reference	26
6.7.1 Detailed Description	27
6.8 RadixSort.h	27
6.9 include/ShellSort.h File Reference	28
6.9.1 Detailed Description	28
6.10 ShellSort.h	29
6.11 include/SortMethod.h File Reference	29
6.11.1 Detailed Description	30
6.12 SortMethod.h	30
6.13 src/FunctionsSort.cc File Reference	31
6.13.1 Detailed Description	32
6.13.2 Function Documentation	33
6.13.2.1 deltasort()	33
6.13.2.2 divide()	33
6.13.2.3 heapify()	34
6.13.2.4 heapSortMethod()	34
6.13.2.5 insertionSortMethod()	35
6.13.2.6 merge()	35
6.13.2.7 mergeSortMethod()	36
6.13.2.8 operator<<()	36
6.13.2.9 radixSortMethod()	37
6.13.2.10 shellSortMethod()	37
6.14 src/main.cc File Reference	38
6.14.1 Detailed Description	38
Index	39

Chapter 1

Sorting algorithms for sequences

1.0.1 Description

Through this program developed in c++17 we can study the different types of sorting algorithms that exist and their complexity. The following algorithms are used:

- [Insertion](#)
- Mergesort
- Shellsort
- Heapsort
- Radixsort

1.0.2 Usage

To compile the program, the make command is used in the main directory and the executable of the program is located in the /bin/main directory.

1.0.3 Author

Fabrizio Daniell Perilli Martín – alu0101138589@ull.edu.es

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

SortMethod< Key >	19
HeapSort< Key >	9
Insertion< Key >	11
MergeSort< Key >	13
RadixSort< Key >	15
ShellSort< Key >	17

Chapter 3

Class Index

3.1 Class List

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

HeapSort< Key >	9
Insertion< Key >	11
MergeSort< Key >	13
RadixSort< Key >	15
ShellSort< Key >	17
SortMethod< Key >	19

Chapter 4

File Index

4.1 File List

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

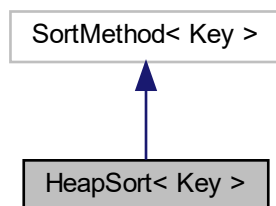
include/ HeapSort.h	
This file contains the HeapSort class	21
include/ Insertion.h	
This file contains the Insertion class	23
include/ MergeSort.h	
This file contains the MergeSort class	24
include/ RadixSort.h	
This file contains the RadixSort class	26
include/ ShellSort.h	
This file contains the ShellSort class	28
include/ SortMethod.h	
This a abstract class that contains the basic methods for the sorting algorithms	29
src/ FunctionsSort.cc	
This file contains the functions that are used in method Sort of the derived classes	31
src/ main.cc	
This file contains the main function of the program	38

Chapter 5

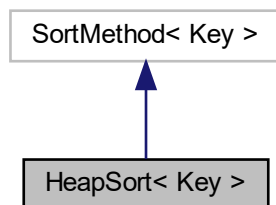
Class Documentation

5.1 HeapSort< Key > Class Template Reference

Inheritance diagram for HeapSort< Key >:



Collaboration diagram for HeapSort< Key >:



Public Member Functions

- [HeapSort](#) (std::vector< Key >, unsigned, bool=false)
Construct a new Heap Sort< Key>:: Heap Sort object.
- void [Sort](#) () override
Sort the sequence using the [HeapSort](#) algorithm.

Additional Inherited Members

5.1.1 Constructor & Destructor Documentation

5.1.1.1 HeapSort()

```
template<class Key >
HeapSort< Key >::HeapSort (
    std::vector< Key > seq,
    unsigned size,
    bool trace = false )
```

Construct a new Heap Sort< Key>:: Heap Sort object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

5.1.2 Member Function Documentation

5.1.2.1 Sort()

```
template<class Key >
void HeapSort< Key >::Sort [override], [virtual]
```

Sort the sequence using the [HeapSort](#) algorithm.

Template Parameters

Key	
-----	--

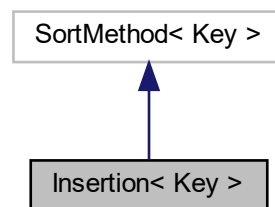
Implements [SortMethod< Key >](#).

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

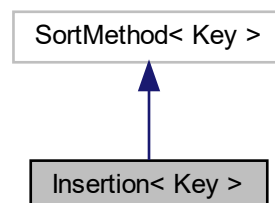
- include/[HeapSort.h](#)

5.2 Insertion< Key > Class Template Reference

Inheritance diagram for Insertion< Key >:



Collaboration diagram for Insertion< Key >:



Public Member Functions

- [Insertion](#) (std::vector< Key >, unsigned, bool=false)
Construct a new Insertion< Key>:: [Insertion](#) object.
- void [Sort](#) () override
Sort the sequence using the [Insertion](#) algorithm.

Additional Inherited Members

5.2.1 Constructor & Destructor Documentation

5.2.1.1 Insertion()

```
template<class Key >
Insertion< Key >::Insertion (
    std::vector< Key > seq,
    unsigned size,
    bool trace = false )
```

Construct a new Insertion< Key>:: [Insertion](#) object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

5.2.2 Member Function Documentation

5.2.2.1 Sort()

```
template<class Key >
void Insertion< Key >::Sort [override], [virtual]
```

Sort the sequence using the [Insertion](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

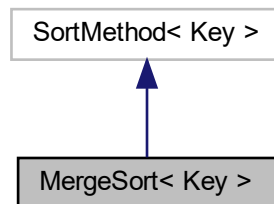
Implements [SortMethod< Key >](#).

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

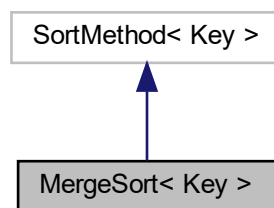
- include/[Insertion.h](#)

5.3 MergeSort< Key > Class Template Reference

Inheritance diagram for MergeSort< Key >:



Collaboration diagram for MergeSort< Key >:



Public Member Functions

- [MergeSort](#) (std::vector< Key >, unsigned, bool=false)
Construct a new Merge Sort< Key>:: Merge Sort object.
- void [Sort](#) () override
Sort the sequence using the [MergeSort](#) algorithm.

Additional Inherited Members

5.3.1 Constructor & Destructor Documentation

5.3.1.1 MergeSort()

```
template<class Key >
MergeSort< Key >::MergeSort (
    std::vector< Key > seq,
    unsigned size,
    bool trace = false )
```

Construct a new Merge Sort< Key>:: Merge Sort object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

5.3.2 Member Function Documentation

5.3.2.1 Sort()

```
template<class Key >
void MergeSort< Key >::Sort [override], [virtual]
```

Sort the sequence using the [MergeSort](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

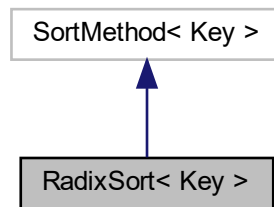
Implements [SortMethod< Key >](#).

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

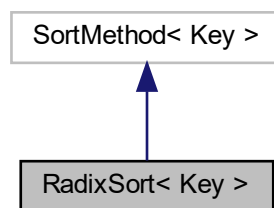
- include/[MergeSort.h](#)

5.4 RadixSort< Key > Class Template Reference

Inheritance diagram for RadixSort< Key >:



Collaboration diagram for RadixSort< Key >:



Public Member Functions

- [RadixSort](#) (std::vector< Key >, unsigned, bool=false)
Construct a new Radix Sort< Key>:: Radix Sort object.
- void [Sort](#) () override
Sort the sequence using the [RadixSort](#) algorithm.

Additional Inherited Members

5.4.1 Constructor & Destructor Documentation

5.4.1.1 RadixSort()

```
template<class Key >
RadixSort< Key >::RadixSort (
    std::vector< Key > seq,
    unsigned size,
    bool trace = false )
```

Construct a new Radix Sort< Key>:: Radix Sort object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

5.4.2 Member Function Documentation

5.4.2.1 Sort()

```
template<class Key >
void RadixSort< Key >::Sort [override], [virtual]
```

Sort the sequence using the [RadixSort](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

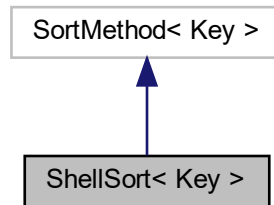
Implements [SortMethod< Key >](#).

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

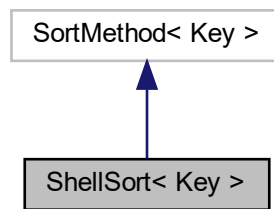
- include/[RadixSort.h](#)

5.5 ShellSort< Key > Class Template Reference

Inheritance diagram for ShellSort< Key >:



Collaboration diagram for ShellSort< Key >:



Public Member Functions

- [ShellSort](#) (std::vector< Key >, unsigned, bool=false)
Construct a new Shell Sort< Key>:: Shell Sort object.
- void [Sort](#) () override
Sort the sequence using the [ShellSort](#) algorithm.

Additional Inherited Members

5.5.1 Constructor & Destructor Documentation

5.5.1.1 ShellSort()

```
template<class Key >
ShellSort< Key >::ShellSort (
    std::vector< Key > seq,
    unsigned size,
    bool trace = false )
```

Construct a new Shell Sort< Key>:: Shell Sort object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

5.5.2 Member Function Documentation

5.5.2.1 Sort()

```
template<class Key >
void ShellSort< Key >::Sort [override], [virtual]
```

Sort the sequence using the [ShellSort](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

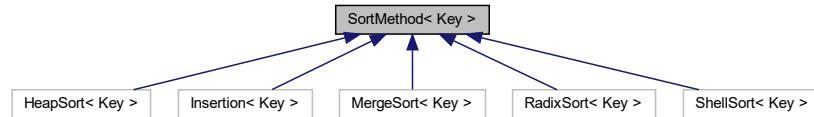
Implements [SortMethod< Key >](#).

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

- include/[ShellSort.h](#)

5.6 SortMethod< Key > Class Template Reference

Inheritance diagram for SortMethod< Key >:



Public Member Functions

- [SortMethod](#) ()
Construct a new Sort Method< Key>:: Sort Method object.
- [SortMethod](#) (std::vector< Key > &, unsigned, bool=false)
Construct a new Sort Method< Key>:: Sort Method object.
- virtual void [Sort](#) ()=0

Protected Attributes

- unsigned **size_**
- std::vector< Key > **seq_**
- bool **trace_**

5.6.1 Constructor & Destructor Documentation

5.6.1.1 SortMethod() [1/2]

```
template<class Key >
SortMethod< Key >::SortMethod
```

Construct a new Sort Method< Key>:: Sort Method object.

Template Parameters

<i>Key</i>	
------------	--

5.6.1.2 SortMethod() [2/2]

```
template<class Key >
SortMethod< Key >::SortMethod (
```

```
std::vector< Key > & seq,
unsigned size,
bool trace = false )
```

Construct a new Sort Method< Key>:: Sort Method object.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

5.6.2 Member Function Documentation

5.6.2.1 Sort()

```
template<class Key >
virtual void SortMethod< Key >::Sort ( ) [pure virtual]
```

Implemented in [HeapSort< Key >](#), [Insertion< Key >](#), [MergeSort< Key >](#), [RadixSort< Key >](#), and [ShellSort< Key >](#).

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

- [include/SortMethod.h](#)

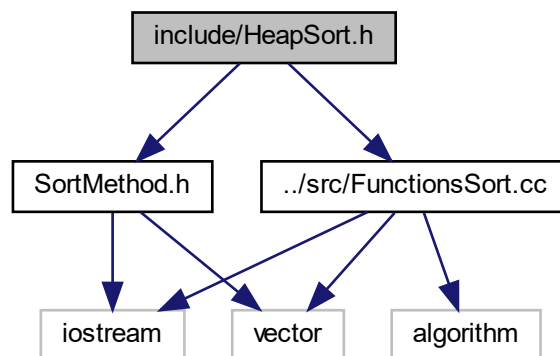
Chapter 6

File Documentation

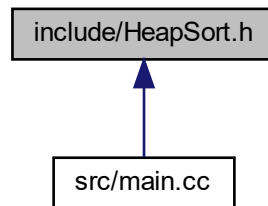
6.1 include/HeapSort.h File Reference

This file contains the [HeapSort](#) class.

```
#include "SortMethod.h"  
#include "../src/FunctionsSort.cc"  
Include dependency graph for HeapSort.h:
```



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



Classes

- class [HeapSort< Key >](#)

6.1.1 Detailed Description

This file contains the [HeapSort](#) class.

Author

Fabrizio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

Copyright (c) 2023

6.2 HeapSort.h

[Go to the documentation of this file.](#)

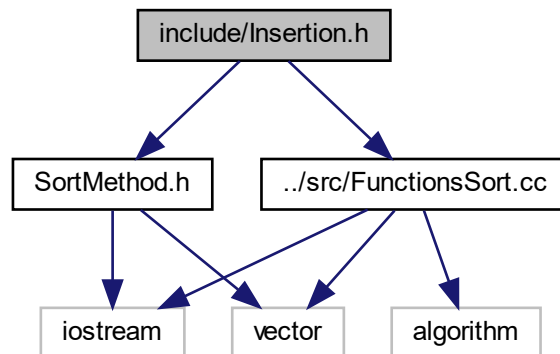
```

1
2 #include "SortMethod.h"
3 #include "../src/FunctionsSort.cc"
4
5
6 template <class Key>
7 class HeapSort : public SortMethod<Key>
8 {
9 public:
10     HeapSort(std::vector<Key>, unsigned, bool = false);
11     void Sort() override;
12 };
13
14 template <class Key>
15 HeapSort<Key>::HeapSort(std::vector<Key> seq, unsigned size, bool trace) : SortMethod<Key>(seq, size,
16     trace) {}
17
18
19 template <class Key>
20 void HeapSort<Key>::Sort()
21 {
22     heapSortMethod(this->seq_, this->size_, this->trace_);
23 }
  
```

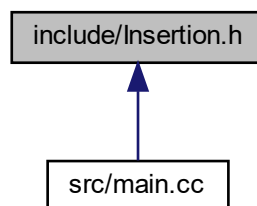
6.3 include/Insertion.h File Reference

This file contains the [Insertion](#) class.

```
#include "SortMethod.h"  
#include "../src/FunctionsSort.cc"  
Include dependency graph for Insertion.h:
```



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



Classes

- class [Insertion](#)< [Key](#) >

6.3.1 Detailed Description

This file contains the [Insertion](#) class.

Author

Fabrizio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

Copyright (c) 2023

6.4 Insertion.h

[Go to the documentation of this file.](#)

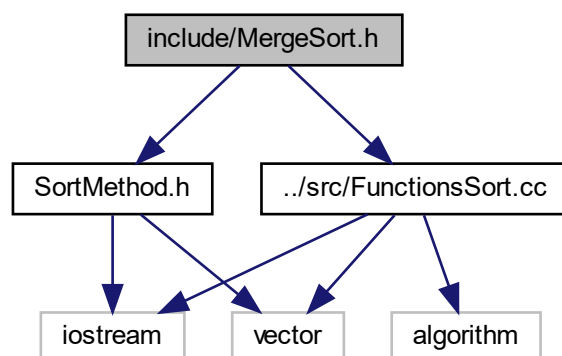
```
1
12 #include "SortMethod.h"
13 #include "../src/FunctionsSort.cc"
14
15
16
17 template <class Key>
18 class Insertion : public SortMethod<Key>
19 {
20 public:
21     Insertion(std::vector<Key>, unsigned, bool = false);
22     void Sort() override;
23 };
24
25
26
27 template <class Key>
28 Insertion<Key>::Insertion(std::vector<Key> seq, unsigned size, bool trace) : SortMethod<Key>(seq, size,
29     trace) {}
30
31
32
33
34
35
36
37 template <class Key>
38 void Insertion<Key>::Sort()
39 {
40     insertionSortMethod(this->seq_, this->size_, this->trace_);
41 }
42 }
```

6.5 include/MergeSort.h File Reference

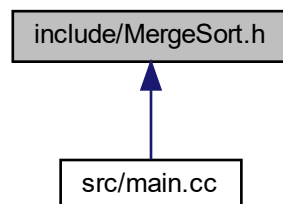
This file contains the [MergeSort](#) class.

```
#include "SortMethod.h"
#include "../src/FunctionsSort.cc"
```

Include dependency graph for MergeSort.h:



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



Classes

- class [MergeSort< Key >](#)

6.5.1 Detailed Description

This file contains the [MergeSort](#) class.

Author

Fabrizio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

Copyright (c) 2023

6.6 MergeSort.h

[Go to the documentation of this file.](#)

```

1
12 #include "SortMethod.h"
13 #include "../src/FunctionsSort.cc"
14
15 template <class Key>
16 class MergeSort : public SortMethod<Key>
17 {
18 public:
19     MergeSort(std::vector<Key>, unsigned, bool = false);
20     void Sort() override;
21 };
22
23
32 template <class Key>
33 MergeSort<Key>::MergeSort(std::vector<Key> seq, unsigned size, bool trace) : SortMethod<Key>(seq, size,
    trace) {}
34
35
41 template <class Key>
42 void MergeSort<Key>::Sort()
43 {
44     mergeSortMethod(this->seq_, this->size_, this->trace_);
45 }

```

6.7 include/RadixSort.h File Reference

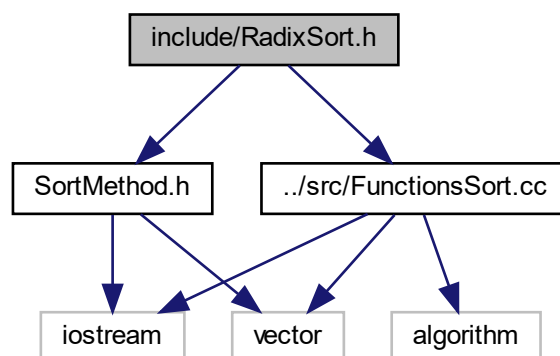
This file contains the [RadixSort](#) class.

```

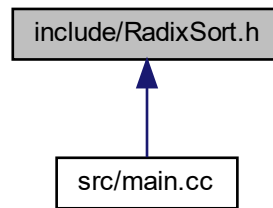
#include "SortMethod.h"
#include "../src/FunctionsSort.cc"

```

Include dependency graph for RadixSort.h:



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



Classes

- class [RadixSort< Key >](#)

6.7.1 Detailed Description

This file contains the [RadixSort](#) class.

Author

Fabrizio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

Copyright (c) 2023

6.8 RadixSort.h

[Go to the documentation of this file.](#)

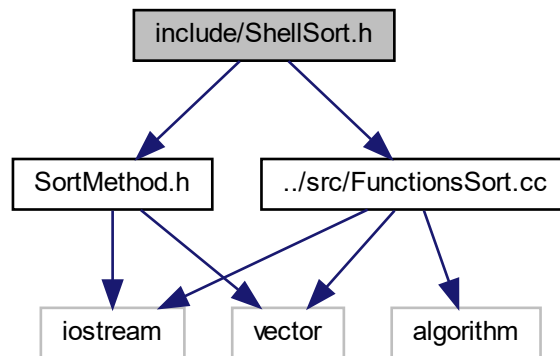
```

1
2 #include "SortMethod.h"
3 #include "../src/FunctionsSort.cc"
4
5 template <class Key>
6 class RadixSort : public SortMethod<Key>
7 {
8 public:
9     RadixSort(std::vector<Key>, unsigned, bool = false);
10    void Sort() override;
11 };
12
13 template <class Key>
14 RadixSort<Key>::RadixSort(std::vector<Key> seq, unsigned size, bool trace) : SortMethod<Key>(seq, size,
15     trace) {}
16
17
18 template <class Key>
19 void RadixSort<Key>::Sort()
20 {
21     radixSortMethod(this->seq_, this->size_, this->trace_);
22 }
  
```

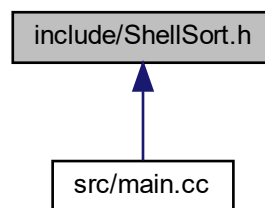
6.9 include/ShellSort.h File Reference

This file contains the [ShellSort](#) class.

```
#include "SortMethod.h"  
#include "../src/FunctionsSort.cc"  
Include dependency graph for ShellSort.h:
```



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



Classes

- class [ShellSort](#)< [Key](#) >

6.9.1 Detailed Description

This file contains the [ShellSort](#) class.

Author

Fabrizio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

Copyright (c) 2023

6.10 ShellSort.h

[Go to the documentation of this file.](#)

```

1
12 #include "SortMethod.h"
13 #include "../src/FunctionsSort.cc"
14
15 template <class Key>
16 class ShellSort : public SortMethod<Key>
17 {
18 public:
19     ShellSort(std::vector<Key>, unsigned, bool = false);
20     void Sort() override;
21 };
22
23 template <class Key>
24 ShellSort<Key>::ShellSort(std::vector<Key> seq, unsigned size, bool trace) : SortMethod<Key>(seq, size,
25     trace) {}
26
27
28
29
30
31 template <class Key>
32 void ShellSort<Key>::Sort()
33 {
34     shellSortMethod(this->seq_, this->size_, this->trace_);
35 }
36

```

6.11 include/SortMethod.h File Reference

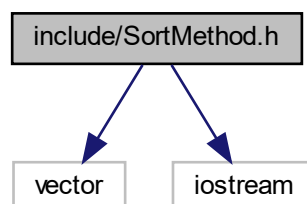
This is an abstract class that contains the basic methods for the sorting algorithms.

```

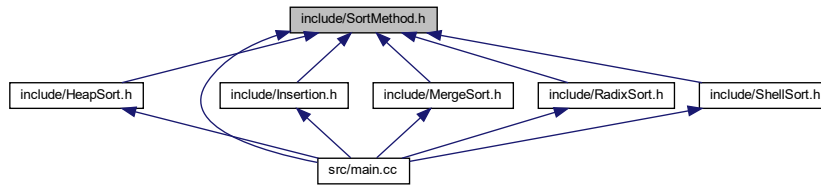
#include <vector>
#include <iostream>

```

Include dependency graph for SortMethod.h:



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



Classes

- class [SortMethod< Key >](#)

6.11.1 Detailed Description

This is an abstract class that contains the basic methods for the sorting algorithms.

Author

Fabrizio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

Copyright (c) 2023

6.12 SortMethod.h

[Go to the documentation of this file.](#)

```

1
12 #pragma once
13 #include <vector>
14 #include <iostream>
15
16 template <class Key>
17 class SortMethod
18 {
19 protected:
20     unsigned size_;
21     std::vector<Key> seq_;
22     bool trace_;
23
24 public:
25     SortMethod();
26     SortMethod(std::vector<Key>&, unsigned, bool = false);
  
```

```

27  virtual void Sort() = 0;
28  };
29
30
36  template <class Key>
37  SortMethod<Key>::SortMethod() {}
38
39
48  template <class Key>
49  SortMethod<Key>::SortMethod(std::vector<Key>& seq, unsigned size, bool trace)
50  {
51      seq_ = seq;
52      size_ = size;
53      trace_ = trace;
54  }

```

6.13 src/FunctionsSort.cc File Reference

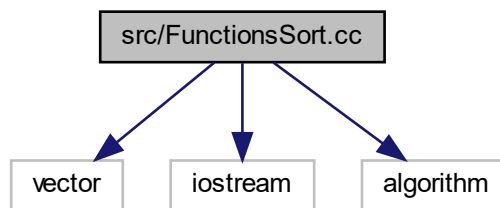
This file contains the functions that are used in method Sort of the derived classes.

```

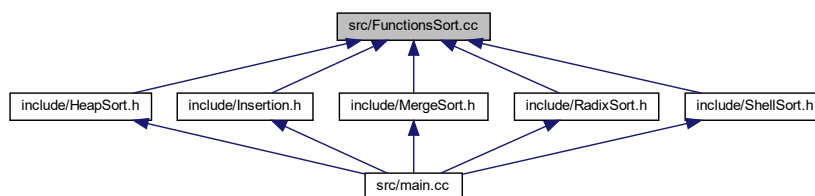
#include <vector>
#include <iostream>
#include <algorithm>

```

Include dependency graph for FunctionsSort.cc:



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



Functions

- `template<class Key >`
void `divide` (std::vector< Key > &seq, int ini, int fin, bool trace, int count)

- Divide the sequence in two parts and call the Merge function.*

 - `template<class Key >`
`void heapify (std::vector< Key > &seq, int size, int i, bool trace, int count)`
Heapify algorithm.
 - `template<class Key >`
`void deltasort (int delta, std::vector< Key > &seq, int size, bool trace, int count)`
Sort the sequence using the deltasort algorithm for the [ShellSort](#) algorithm.
 - `template<class Key >`
`std::ostream & operator<< (std::ostream &os, const std::vector< Key > &seq)`
Overload the operator << to print the sequence.
 - `template<class Key >`
`void insertionSortMethod (std::vector< Key > &seq, unsigned size, bool trace=false)`
Sort the sequence using the [Insertion](#) algorithm.
 - `template<class Key >`
`void merge (std::vector< Key > &seq, int ini, int mid, int fin, bool trace, int count)`
Merge the two parts of the sequence.
 - `template<class Key >`
`void mergeSortMethod (std::vector< Key > &seq, unsigned size, bool trace)`
Sort the sequence using the [MergeSort](#) algorithm.
 - `template<class Key >`
`void heapSortMethod (std::vector< Key > &seq, unsigned size, bool trace)`
Heapsort algorithm.
 - `template<class Key >`
`void radixSortMethod (std::vector< Key > &seq, unsigned size, bool trace)`
Sort the sequence using the [RadixSort](#) algorithm.
 - `template<class Key >`
`void shellSortMethod (std::vector< Key > &seq, int size, bool trace)`
Sort the sequence using the [ShellSort](#) algorithm.

6.13.1 Detailed Description

This file contains the functions that are used in method Sort of the derived classes.

Author

Fabrizio Daniell Perilli Martin

Version

0.1

Date

2023-04-11

Copyright

Copyright (c) 2023

6.13.2 Function Documentation

6.13.2.1 deltasort()

```
template<class Key >
void deltasort (
    int delta,
    std::vector< Key > & seq,
    int size,
    bool trace,
    int count )
```

Sort the sequence using the deltasort algorithm for the [ShellSort](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>delta</i>	
<i>seq</i>	
<i>size</i>	
<i>trace</i>	
<i>count</i>	

6.13.2.2 divide()

```
template<class Key >
void divide (
    std::vector< Key > & seq,
    int ini,
    int fin,
    bool trace,
    int count )
```

Divide the sequence in two parts and call the Merge function.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>ini</i>	

Parameters

<i>fin</i>	
<i>trace</i>	
<i>count</i>	

6.13.2.3 heapify()

```
template<class Key >
void heapify (
    std::vector< Key > & seq,
    int size,
    int i,
    bool trace,
    int count )
```

Heapify algorithm.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>i</i>	
<i>trace</i>	

6.13.2.4 heapSortMethod()

```
template<class Key >
void heapSortMethod (
    std::vector< Key > & seq,
    unsigned size,
    bool trace )
```

Heapsort algorithm.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

6.13.2.5 insertionSortMethod()

```
template<class Key >
void insertionSortMethod (
    std::vector< Key > & seq,
    unsigned size,
    bool trace = false )
```

Sort the sequence using the [Insertion](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

6.13.2.6 merge()

```
template<class Key >
void merge (
    std::vector< Key > & seq,
    int ini,
    int mid,
    int fin,
    bool trace,
    int count )
```

Merge the two parts of the sequence.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>ini</i>	
<i>mid</i>	
<i>fin</i>	
<i>trace</i>	
<i>count</i>	

6.13.2.7 mergeSortMethod()

```
template<class Key >
void mergeSortMethod (
    std::vector< Key > & seq,
    unsigned size,
    bool trace )
```

Sort the sequence using the [MergeSort](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

6.13.2.8 operator<<()

```
template<class Key >
std::ostream & operator<< (
    std::ostream & os,
    const std::vector< Key > & seq )
```

Overload the operator << to print the sequence.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>os</i>	
-----------	--

Parameters

<i>seq</i>	
------------	--

Returns

std::ostream&

6.13.2.9 radixSortMethod()

```
template<class Key >
void radixSortMethod (
    std::vector< Key > & seq,
    unsigned size,
    bool trace )
```

Sort the sequence using the [RadixSort](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
<i>size</i>	
<i>trace</i>	

6.13.2.10 shellSortMethod()

```
template<class Key >
void shellSortMethod (
    std::vector< Key > & seq,
    int size,
    bool trace )
```

Sort the sequence using the [ShellSort](#) algorithm.

Template Parameters

<i>Key</i>	
------------	--

Parameters

<i>seq</i>	
------------	--

Parameters

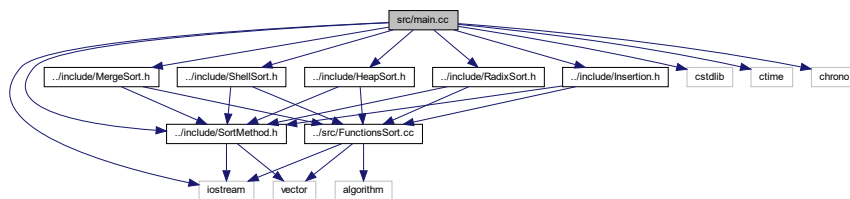
<i>size</i>	
<i>alfa</i>	
<i>trace</i>	

6.14 src/main.cc File Reference

This file contains the main function of the program.

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <chrono>
#include "../include/SortMethod.h"
#include "../include/Insertion.h"
#include "../include/MergeSort.h"
#include "../include/ShellSort.h"
#include "../include/HeapSort.h"
#include "../include/RadixSort.h"
```

Include dependency graph for main.cc:



Functions

- `int main ()`

6.14.1 Detailed Description

This file contains the main function of the program.

Author

Fabrizio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

Copyright (c) 2023

Index

deltasort
 FunctionsSort.cc, [33](#)

divide
 FunctionsSort.cc, [33](#)

FunctionsSort.cc
 deltasort, [33](#)
 divide, [33](#)
 heapify, [34](#)
 heapSortMethod, [34](#)
 insertionSortMethod, [35](#)
 merge, [35](#)
 mergeSortMethod, [36](#)
 operator<<, [36](#)
 radixSortMethod, [37](#)
 shellSortMethod, [37](#)

heapify
 FunctionsSort.cc, [34](#)

HeapSort
 HeapSort< Key >, [10](#)

HeapSort< Key >, [9](#)
 HeapSort, [10](#)
 Sort, [10](#)

heapSortMethod
 FunctionsSort.cc, [34](#)

include/HeapSort.h, [21](#), [22](#)
include/Insertion.h, [23](#), [24](#)
include/MergeSort.h, [24](#), [26](#)
include/RadixSort.h, [26](#), [27](#)
include/ShellSort.h, [28](#), [29](#)
include/SortMethod.h, [29](#), [30](#)

Insertion
 Insertion< Key >, [12](#)

Insertion< Key >, [11](#)
 Insertion, [12](#)
 Sort, [12](#)

insertionSortMethod
 FunctionsSort.cc, [35](#)

merge
 FunctionsSort.cc, [35](#)

MergeSort
 MergeSort< Key >, [13](#)

MergeSort< Key >, [13](#)
 MergeSort, [13](#)
 Sort, [14](#)

mergeSortMethod
 FunctionsSort.cc, [36](#)

operator<<
 FunctionsSort.cc, [36](#)

RadixSort
 RadixSort< Key >, [15](#)

RadixSort< Key >, [15](#)
 RadixSort, [15](#)
 Sort, [16](#)

radixSortMethod
 FunctionsSort.cc, [37](#)

ShellSort
 ShellSort< Key >, [17](#)

ShellSort< Key >, [17](#)
 ShellSort, [17](#)
 Sort, [18](#)

shellSortMethod
 FunctionsSort.cc, [37](#)

Sort
 HeapSort< Key >, [10](#)
 Insertion< Key >, [12](#)
 MergeSort< Key >, [14](#)
 RadixSort< Key >, [16](#)
 ShellSort< Key >, [18](#)
 SortMethod< Key >, [20](#)

SortMethod
 SortMethod< Key >, [19](#)

SortMethod< Key >, [19](#)
 Sort, [20](#)
 SortMethod, [19](#)

src/FunctionsSort.cc, [31](#)
src/main.cc, [38](#)