

## Sort Methods

Generated by Doxygen 1.9.3



<b>1 Sorting algorithms for sequences</b>	<b>1</b>
1.0.1 Description	1
1.0.2 Usage	1
1.0.3 Author	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Class Documentation</b>	<b>9</b>
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	15
5.3.2.1 Sort()	15
5.4 RadixSort< Key > Class Template Reference	15
5.4.1 Constructor & Destructor Documentation	16
5.4.1.1 RadixSort()	16
5.4.2 Member Function Documentation	17
5.4.2.1 Sort()	17
5.5 ShellSort< Key > Class Template Reference	17
5.5.1 Constructor & Destructor Documentation	18
5.5.1.1 ShellSort()	18
5.5.2 Member Function Documentation	19
5.5.2.1 Sort()	19
5.6 SortMethod< Key > Class Template Reference	19
5.6.1 Constructor & Destructor Documentation	20
5.6.1.1 SortMethod()	20
5.6.2 Member Function Documentation	20

---

5.6.2.1 Sort()	20
5.6.2.2 Write()	20
<b>6 File Documentation</b>	<b>23</b>
6.1 include/HeapSort.h File Reference	23
6.1.1 Detailed Description	24
6.2 HeapSort.h	24
6.3 include/Insertion.h File Reference	25
6.3.1 Detailed Description	26
6.4 Insertion.h	27
6.5 include/MergeSort.h File Reference	27
6.5.1 Detailed Description	28
6.6 MergeSort.h	29
6.7 include/RadixSort.h File Reference	30
6.7.1 Detailed Description	31
6.8 RadixSort.h	31
6.9 include/ShellSort.h File Reference	32
6.9.1 Detailed Description	32
6.10 ShellSort.h	33
6.11 include/SortMethod.h File Reference	34
6.11.1 Detailed Description	34
6.12 SortMethod.h	35
6.13 src/main.cc File Reference	35
6.13.1 Detailed Description	36
<b>Index</b>	<b>37</b>

# Chapter 1

## Sorting algorithms for sequences

### 1.0.1 Description

Through this program developed in c++ 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](mailto: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:

<a href="#">HeapSort&lt; Key &gt;</a>	9
<a href="#">Insertion&lt; Key &gt;</a>	11
<a href="#">MergeSort&lt; Key &gt;</a>	13
<a href="#">RadixSort&lt; Key &gt;</a>	15
<a href="#">ShellSort&lt; Key &gt;</a>	17
<a href="#">SortMethod&lt; Key &gt;</a>	19



## Chapter 4

# File Index

### 4.1 File List

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

include/ <a href="#">HeapSort.h</a>	
This file contains the <a href="#">HeapSort</a> class . . . . .	23
include/ <a href="#">Insertion.h</a>	
This file contains the <a href="#">Insertion</a> class . . . . .	25
include/ <a href="#">MergeSort.h</a>	
This file contains the <a href="#">MergeSort</a> class . . . . .	27
include/ <a href="#">RadixSort.h</a>	
This file contains the <a href="#">RadixSort</a> class . . . . .	30
include/ <a href="#">ShellSort.h</a>	
This file contains the <a href="#">ShellSort</a> class . . . . .	32
include/ <a href="#">SortMethod.h</a>	
This a abstract class that contains the basic methods for the sorting algorithms . . . . .	34
src/ <a href="#">main.cc</a>	
This file contains the main function of the program . . . . .	35

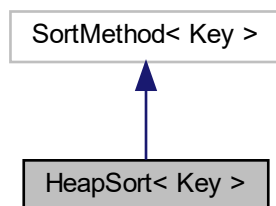


## Chapter 5

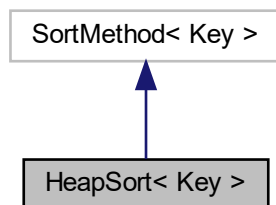
# Class Documentation

### 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)  
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 )
```

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

##### Template Parameters

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

##### Parameters

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

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

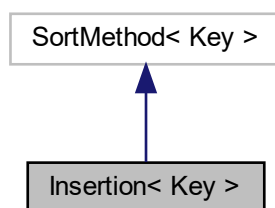
Implements [SortMethod< Key >](#).

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

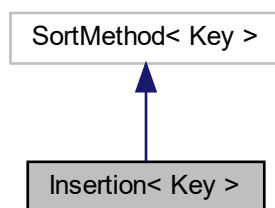
- include/[HeapSort.h](#)

## 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)  
*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 )
```

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

##### Template Parameters

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

##### Parameters

<i>seq</i>	
<i>size</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 >](#).

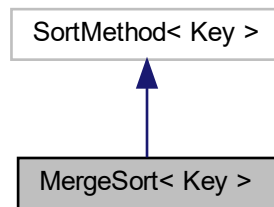
The documentation for this class was generated from the following file:

- [include/Insertion.h](#)

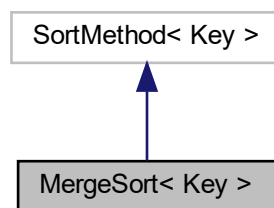


## 5.3 MergeSort< Key > Class Template Reference

Inheritance diagram for MergeSort< Key >:



Collaboration diagram for MergeSort< Key >:



### Public Member Functions

- [MergeSort](#) (std::vector< Key >, unsigned)  
*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 )
```

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

## Template Parameters

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

## Parameters

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

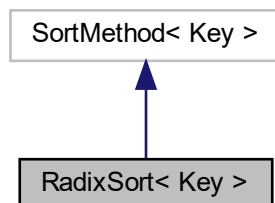
Implements [SortMethod< Key >](#).

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

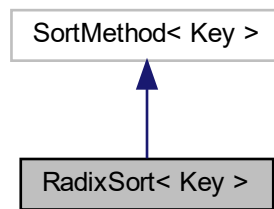
- include/[MergeSort.h](#)

## 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)  
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 )
  
```

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

#### Template Parameters

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

#### Parameters

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

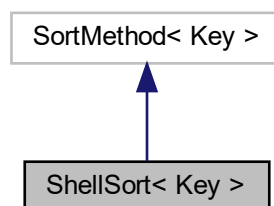
Implements [SortMethod< Key >](#).

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

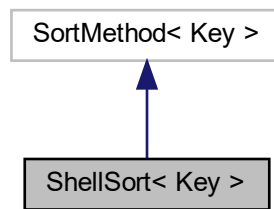
- [include/RadixSort.h](#)

## 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)  
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 )
  
```

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

#### Template Parameters

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

#### Parameters

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

Key	
-----	--

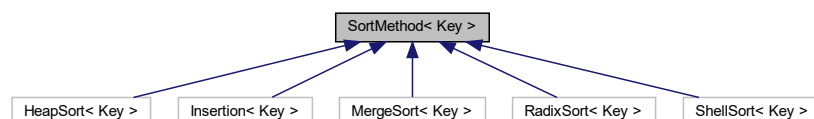
Implements [SortMethod< Key >](#).

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](#) (std::vector< Key >, unsigned)  
*Construct a new Sort Method< Key>:: Sort Method object.*
- virtual void [Sort](#) ()=0
- void [Write](#) (int)  
*Write the sequence in the console.*

### Protected Attributes

- unsigned **size\_**
- std::vector< Key > **seq\_**

## 5.6.1 Constructor & Destructor Documentation

### 5.6.1.1 SortMethod()

```
template<class Key >
SortMethod< Key >::SortMethod (
    std::vector< Key > seq,
    unsigned size )
```

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

#### Template Parameters

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

#### Parameters

<i>seq</i>	
<i>size</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 >](#).

### 5.6.2.2 Write()

```
template<class Key >
void SortMethod< Key >::Write (
    int count )
```

Write the sequence in the console.

#### Template Parameters

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



## Parameters

<i>count</i>	
--------------	--

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

- [include/SortMethod.h](#)



## Chapter 6

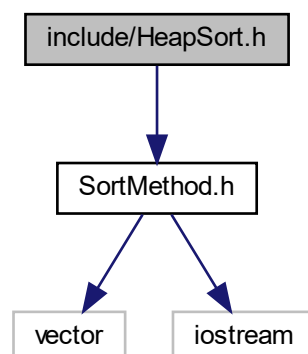
# File Documentation

### 6.1 include/HeapSort.h File Reference

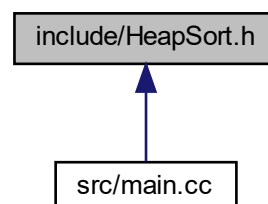
This file contains the [HeapSort](#) class.

```
#include "SortMethod.h"
```

Include 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](mailto: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
12 #include "SortMethod.h"
13
14 template <class Key>
15 class HeapSort : public SortMethod<Key>
16 {
17 public:
18     HeapSort(std::vector<Key>, unsigned);
19     void Sort() override;
20 };
21
22 template <class Key>
23 HeapSort<Key>::HeapSort(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}
24
25 template <class Key>
26 void HeapSort<Key>::Sort()
27 {
28     int count = 0;
29
30     for (size_t i = 1; i < this->size_; i++)
31     {
32         int pos = i;
33         while (pos > 0)
34         {
35             int father = (pos - 1) / 2;
36             if (this->seq_[pos] > this->seq_[father])
37             {
38                 std::swap(this->seq_[pos], this->seq_[father]);
39                 pos = father;
40             }
41             else
42                 break;
43         }
44         this->Write(count);
45         count++;
46     }
47 }
```

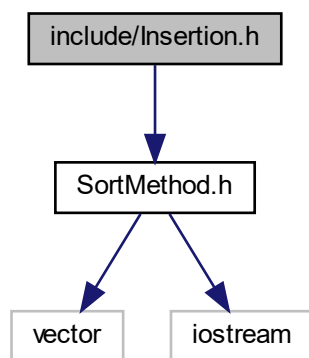
```
60  for (int i = this->size_ - 1; i > 0; i--)
61  {
62      std::swap(this->seq_[0], this->seq_[i]);
63      int pos = 0;
64      int son = 2 * pos + 1;
65      while (son < i)
66      {
67          if (son + 1 < i)
68          {
69              if (this->seq_[son] < this->seq_[son + 1])
70                  son++;
71          }
72          if (this->seq_[pos] < this->seq_[son])
73          {
74              std::swap(this->seq_[pos], this->seq_[son]);
75              pos = son;
76              son = 2 * pos + 1;
77          }
78          else
79              break;
80      }
81      this->Write(count);
82      count++;
83  }
84 }
```

## 6.3 include/Insertion.h File Reference

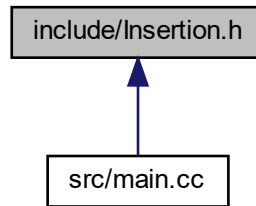
This file contains the [Insertion](#) class.

```
#include "SortMethod.h"
```

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](mailto: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
14 template <class Key>
15 class Insertion : public SortMethod<Key>
16 {
17 public:
18     Insertion(std::vector<Key>, unsigned);
19     void Sort() override;
20 };
21
29 template <class Key>
30 Insertion<Key>::Insertion(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}
31
32
38 template <class Key>
39 void Insertion<Key>::Sort()
40 {
41     int count = 0;
42     int pos, aux;
43     for (size_t i = 0; i < this->size_; i++)
44     {
45         pos = i;
46         aux = this->seq_[i];
47         while (pos > 0 && aux < this->seq_[pos - 1])
48         {
49             std::swap(this->seq_[pos], this->seq_[pos - 1]);
50             pos--;
51         }
52         this->seq_[pos] = aux;
53         this->Write(count);
54         count++;
55     }
56 }

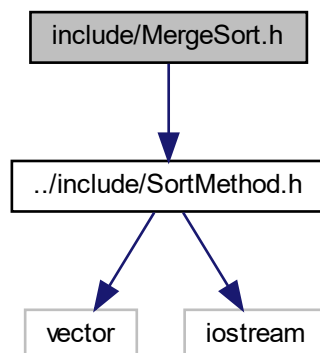
```

## 6.5 include/MergeSort.h File Reference

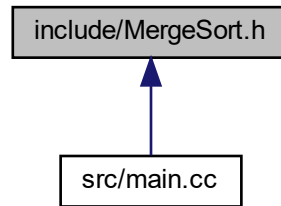
This file contains the [MergeSort](#) class.

```
#include "../include/SortMethod.h"
```

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](mailto: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 "../include/SortMethod.h"
13
14 template <class Key>
15 class MergeSort : public SortMethod<Key>
16 {
17 public:
18     MergeSort(std::vector<Key>, unsigned);
19     void Sort() override;
20
21 private:
22     void Merge(int, int, int);
23     void Divide(int, int);
24     int count = 0;
25 };
26
27 template <class Key>
28 MergeSort<Key>::MergeSort(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}
29
30 template <class Key>
31 void MergeSort<Key>::Sort()
32 {
33     Divide(0, this->size_ - 1);
34     this->Write(count);
35     count++;
36 }
37
38 template <class Key>
39 void MergeSort<Key>::Divide(int ini, int fin)
40 {
41     if (ini < fin)
42     {
43         int middle = ini + (fin - ini) / 2;
44         Divide(ini, middle);
45         Divide(middle + 1, fin);
46         Merge(ini, middle, fin);
47         this->Write(count);
48         count++;
49     }
50 }
51
52 template <class Key>
53 void MergeSort<Key>::Merge(int ini, int middle, int fin)
54 {
55     int i, j, k;
56     int n1 = middle - ini + 1;
57     int n2 = fin - middle;
58
59     std::vector<Key> L(n1), R(n2);
60
61     for (i = 0; i < n1; i++)
62         L[i] = this->seq_[ini + i];
63     for (j = 0; j < n2; j++)
64         R[j] = this->seq_[middle + 1 + j];
65
66     i = 0;
67     j = 0;
68     k = ini;
69
70     while (i < n1 && j < n2)
71     {
72         if (L[i] <= R[j])
73         {
74             this->seq_[k] = L[i];
75             i++;
76         }
77         else
78         {
79             this->seq_[k] = R[j];
80             j++;
81         }
82         k++;
83     }
84
85     while (i < n1)
86     {
87         this->seq_[k] = L[i];
88         i++;
89     }
90 }

```

```
120     k++;
121 }
122
123 while (j < n2)
124 {
125     this->seq_[k] = R[j];
126     j++;
127     k++;
128 }
129
130 this->Write(count);
131 count++;
132 }
```

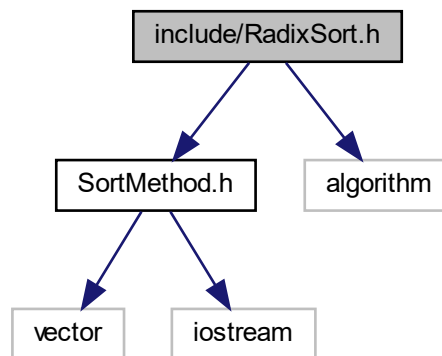
## 6.7 include/RadixSort.h File Reference

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

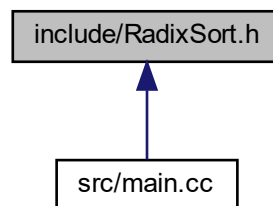
```
#include "SortMethod.h"
```

```
#include <algorithm>
```

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](mailto: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.](#)

```

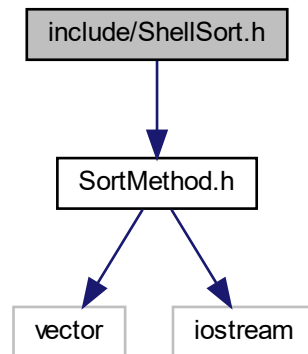
1
2 #include "SortMethod.h"
3 #include <algorithm>
4
5 template <class Key>
6 class RadixSort : public SortMethod<Key>
7 {
8 public:
9     RadixSort(std::vector<Key>, unsigned);
10    void Sort() override;
11 };
12
13 template <class Key>
14 RadixSort<Key>::RadixSort(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}
15
16 template <class Key>
17 void RadixSort<Key>::Sort()
18 {
19     int count = 0;
20     auto max = *std::max_element(this->seq_.begin(), this->seq_.end());
21
22     for (int exp = 1; max / exp > 0; exp *= 10)
23     {
24         int output[this->size_];
25         int i, bucket[10] = {0};
26
27         for (i = 0; i < (int)this->size_; i++)
28             bucket[(this->seq_[i] / exp) % 10]++;
29
30         for (i = 1; i < 10; i++)
31             bucket[i] += bucket[i - 1];
32
33         for (i = this->size_ - 1; i >= 0; i--)
34         {
35             output[bucket[(this->seq_[i] / exp) % 10] - 1] = this->seq_[i];
36             bucket[(this->seq_[i] / exp) % 10]--;
37         }
38
39         for (i = 0; i < (int)this->size_; i++)
40             this->seq_[i] = output[i];
41
42         this->Write(count);
43         count++;
44     }
45 }
```

## 6.9 include/ShellSort.h File Reference

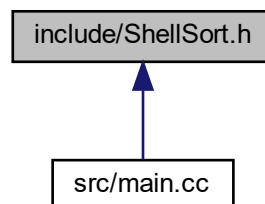
This file contains the [ShellSort](#) class.

```
#include "SortMethod.h"
```

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](mailto: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.](#)

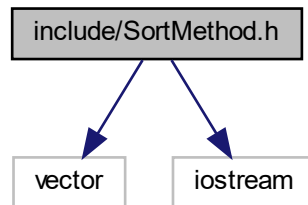
```
1
12 #include "SortMethod.h"
13
14 template <class Key>
15 class ShellSort : public SortMethod<Key>
16 {
17 public:
18     ShellSort(std::vector<Key>, unsigned);
19     void Sort() override;
20 };
21
22 template <class Key>
23 ShellSort<Key>::ShellSort(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}
24
25 template <class Key>
26 void ShellSort<Key>::Sort()
27 {
28     int count = 0;
29     int alfa = this->size_ / 2;
30     int aux, pos;
31     std::cout << "Alfa: " << alfa << std::endl;
32     while (alfa > 0)
33     {
34         for (size_t i = alfa; i < this->size_; i++)
35         {
36             pos = i;
37             aux = this->seq_[i];
38             while (pos >= alfa && aux < this->seq_[pos - alfa])
39             {
40                 std::swap(this->seq_[pos], this->seq_[pos - alfa]);
41                 pos -= alfa;
42             }
43             this->Write(count);
44             count++;
45         }
46         alfa /= 2;
47         if (alfa > 0)
48             std::cout << "Alfa: " << alfa << std::endl;
49     }
50 }
```

## 6.11 include/SortMethod.h File Reference

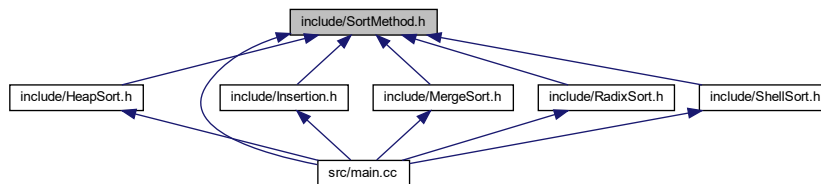
This is an abstract class that contains the basic methods for the sorting algorithms.

```
#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](mailto: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 #include <vector>
13 #include <iostream>
14 #pragma once
15
16 template <class Key>
17 class SortMethod
18 {
19 protected:
20     unsigned size_;
21     std::vector<Key> seq_;
22
23 public:
24     SortMethod() {}
25     SortMethod(std::vector<Key>, unsigned);
26     virtual void Sort() = 0;
27     void Write(int);
28 };
29
30
31 template <class Key>
32 SortMethod<Key>::SortMethod(std::vector<Key> seq, unsigned size)
33 {
34     seq_ = seq;
35     size_ = size;
36 }
37
38 template <class Key>
39 void SortMethod<Key>::Write(int count)
40 {
41     std::cout << "i: " << count << " -->";
42     for (unsigned i = 0; i < size_; i++)
43     {
44         if (i == 0)
45             std::cout << " | ";
46         std::cout << seq_[i] << " | ";
47     }
48     std::cout << std::endl;
49 }

```

## 6.13 src/main.cc File Reference

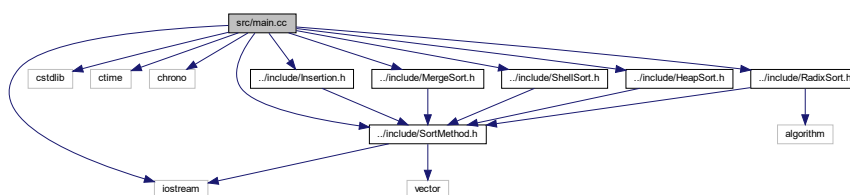
This file contains the main function of the program.

```

#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 ()`

### 6.13.1 Detailed Description

This file contains the main function of the program.

#### Author

Fabrizio Daniell Perilli Martin [alu0101138589@ull.edu.es](mailto:alu0101138589@ull.edu.es)

#### Version

0.1

#### Date

2023-04-03

#### Copyright

Copyright (c) 2023



# Index

HeapSort  
    HeapSort< Key >, [10](#)  
HeapSort< Key >, [9](#)  
    HeapSort, [10](#)  
    Sort, [10](#)  
  
include/HeapSort.h, [23](#), [24](#)  
include/Insertion.h, [25](#), [27](#)  
include/MergeSort.h, [27](#), [29](#)  
include/RadixSort.h, [30](#), [31](#)  
include/ShellSort.h, [32](#), [33](#)  
include/SortMethod.h, [34](#), [35](#)  
Insertion  
    Insertion< Key >, [12](#)  
Insertion< Key >, [11](#)  
    Insertion, [12](#)  
    Sort, [12](#)  
  
MergeSort  
    MergeSort< Key >, [13](#)  
MergeSort< Key >, [13](#)  
    MergeSort, [13](#)  
    Sort, [15](#)  
  
RadixSort  
    RadixSort< Key >, [16](#)  
RadixSort< Key >, [15](#)  
    RadixSort, [16](#)  
    Sort, [17](#)  
  
ShellSort  
    ShellSort< Key >, [18](#)  
ShellSort< Key >, [17](#)  
    ShellSort, [18](#)  
    Sort, [19](#)  
Sort  
    HeapSort< Key >, [10](#)  
    Insertion< Key >, [12](#)  
    MergeSort< Key >, [15](#)  
    RadixSort< Key >, [17](#)  
    ShellSort< Key >, [19](#)  
    SortMethod< Key >, [20](#)  
SortMethod  
    SortMethod< Key >, [20](#)  
SortMethod< Key >, [19](#)  
    Sort, [20](#)  
    SortMethod, [20](#)  
    Write, [20](#)  
src/main.cc, [35](#)  
  
Write