Sort Methods

Generated by Doxygen 1.9.3

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

Fabrizzio 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:

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

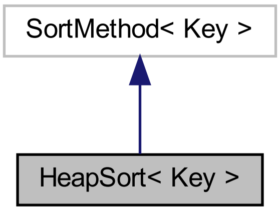
Here is a list of all documented files with brief descriptions:

Chapter 5

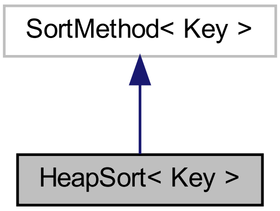
Class Documentation

5.1 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

Key	

Parameters

seq	
size	

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	

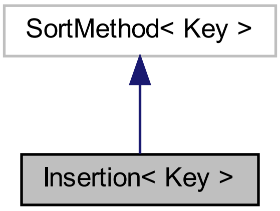
Implements SortMethod< Key >.

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

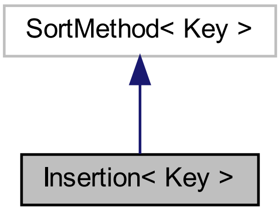
- include/HeapSort.h

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)

    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

Key	

Parameters

seq	
size	

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

Key	

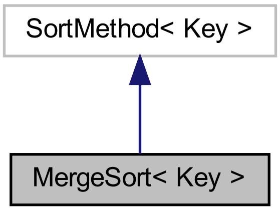
Implements SortMethod< Key >.

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

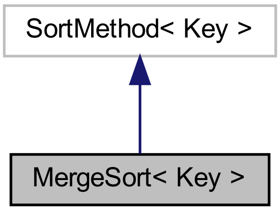
- include/Insertion.h

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

Key	

Parameters

seq	
size	

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

Key	

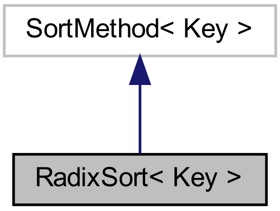
Implements SortMethod< Key >.

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

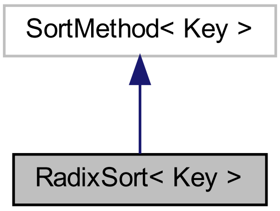
- include/MergeSort.h

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

Key	

Parameters

seq	
size	

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

Key	

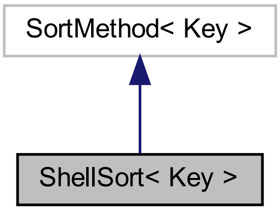
Implements SortMethod< Key >.

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

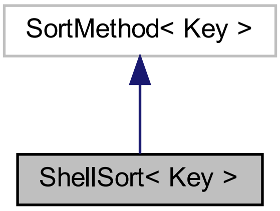
- include/RadixSort.h

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

Key	

Parameters

seq	
size	

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	

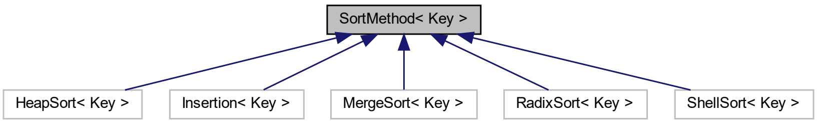
Implements SortMethod< Key >.

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

Key	

Parameters

seq	
size	

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

Key	

Parameters

count	

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

- include/SortMethod.h

Chapter 6

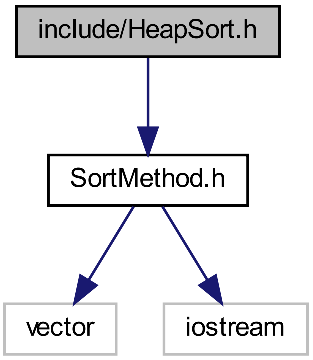
File Documentation

6.1 include/HeapSort.h File Reference

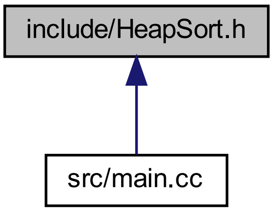
This file contains the HeapSort class.

```
#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

Fabrizzio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

6.2 HeapSort.h

Go to the documentation of this file.

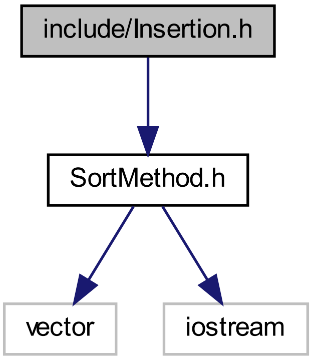
```

#include "SortMethod.h"

template <class Key>
class HeapSort : public SortMethod<Key>
{
public:
  HeapSort(std::vector<Key>, unsigned);
  void Sort() override;
};

template <class Key>
HeapSort<Key>::HeapSort(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}


template <class Key>
void HeapSort<Key>::Sort()
{
  int count = 0;

  for (size_t i = 1; i < this->size_; i++)
  {
    int pos = i;
    while (pos > 0)
    {
      int father = (pos - 1) / 2;
      if (this->seq_[pos] > this->seq_[father])
      {
        std::swap(this->seq_[pos], this->seq_[father]);
        pos = father;
      }
      else
        break;
    }
    this->Write(count);
    count++;
  }
  for (int i = this->size_ - 1; i > 0; i--)
  {
    std::swap(this->seq_[0], this->seq_[i]);
    int pos = 0;
    int son = 2 * pos + 1;
    while (son < i)
    {
      if (son + 1 < i)
      {
        if (this->seq_[son] < this->seq_[son + 1])
          son++;
      }
      if (this->seq_[pos] < this->seq_[son])
      {
        std::swap(this->seq_[pos], this->seq_[son]);
        pos = son;
        son = 2 * pos + 1;
      }
      else
        break;
    }
    this->Write(count);
    count++;
  }
}
```

6.3 include/Insertion.h File Reference

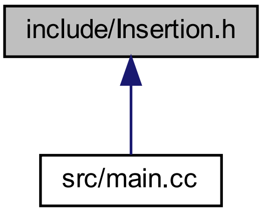
This file contains the Insertion class.

`#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

Fabrizzio 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.

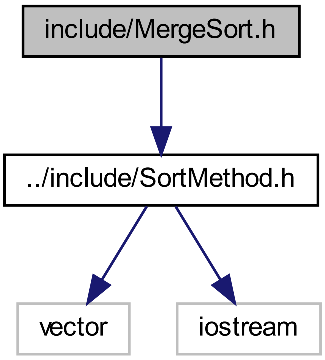
```

#include "SortMethod.h"

template <class Key>
class Insertion : public SortMethod<Key>
{
public:
  Insertion(std::vector<Key>, unsigned);
  void Sort() override;
};

template <class Key>
Insertion<Key>::Insertion(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}


template <class Key>
void Insertion<Key>::Sort()
{
  int count = 0;
  int pos, aux;
  for (size_t i = 0; i < this->size_; i++)
  {
    pos = i;
    aux = this->seq_[i];
    while (pos > 0 && aux < this->seq_[pos - 1])
    {
      std::swap(this->seq_[pos], this->seq_[pos - 1]);
      pos--;
    }
    this->seq_[pos] = aux;
    this->Write(count);
    count++;
  }
}
```

6.5 include/MergeSort.h File Reference

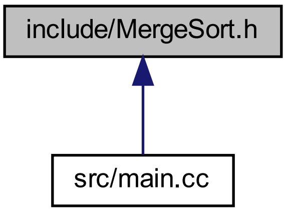
This file contains the MergeSort class.

```
#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

Fabrizzio 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.

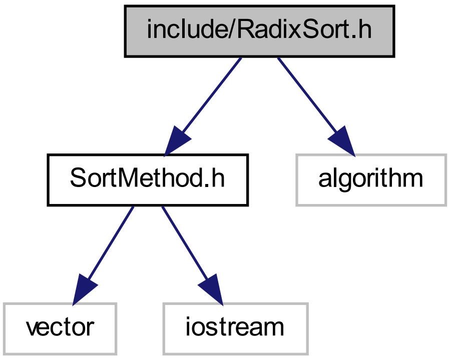
```

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

template <class Key>
class MergeSort : public SortMethod<Key>
{
public:
  MergeSort(std::vector<Key>, unsigned);
  void Sort() override;

private:
  void Merge(int, int, int);
  void Divide(int, int);
  int count = 0;
};

template <class Key>
MergeSort<Key>::MergeSort(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}


template <class Key>
void MergeSort<Key>::Sort()
{
  Divide(0, this->size_ - 1);
  this->Write(count);
  count++;
}


template <class Key>
void MergeSort<Key>::Divide(int ini, int fin)
{

  if (ini < fin)
  {
    int middle = ini + (fin - ini) / 2;
    Divide(ini, middle);
    Divide(middle + 1, fin);
    Merge(ini, middle, fin);
    this->Write(count);
    count++;
  }
}


template <class Key>
void MergeSort<Key>::Merge(int ini, int middle, int fin)
{
  int i, j, k;
  int n1 = middle - ini + 1;
  int n2 = fin - middle;

  std::vector<Key> L(n1), R(n2);

  for (i = 0; i < n1; i++)
    L[i] = this->seq_[ini + i];
  for (j = 0; j < n2; j++)
    R[j] = this->seq_[middle + 1 + j];

  i = 0;
  j = 0;
  k = ini;

  while (i < n1 && j < n2)
  {
    if (L[i] <= R[j])
    {
      this->seq_[k] = L[i];
      i++;
    }
    else
    {
      this->seq_[k] = R[j];
      j++;
    }
    k++;
  }

  while (i < n1)
  {
    this->seq_[k] = L[i];
    i++;
    k++;
  }

  while (j < n2)
  {
    this->seq_[k] = R[j];
    j++;
    k++;
  }

  this->Write(count);
  count++;
}
```

6.7 include/RadixSort.h File Reference

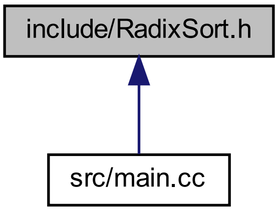
This file contains the RadixSort class.

```
#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

Fabrizzio 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.

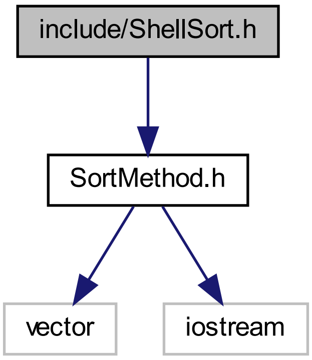
```
#include "SortMethod.h"
#include <algorithm>

template <class Key>
class RadixSort : public SortMethod<Key>
{
public:
  RadixSort(std::vector<Key>, unsigned);
  void Sort() override;
};


template <class Key>
RadixSort<Key>::RadixSort(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}


template <class Key>
void RadixSort<Key>::Sort()
{
  int count = 0;
  auto max = *std::max_element(this->seq_.begin(), this->seq_.end());

  for (int exp = 1; max / exp > 0; exp *= 10)
  {
    int output[this->size_];
    int i, bucket[10] = {0};

    for (i = 0; i < (int)this->size_; i++)
      bucket[(this->seq_[i] / exp) % 10]++;

    for (i = 1; i < 10; i++)
      bucket[i] += bucket[i - 1];

    for (i = this->size_ - 1; i >= 0; i--)
    {
      output[bucket[(this->seq_[i] / exp) % 10] - 1] = this->seq_[i];
      bucket[(this->seq_[i] / exp) % 10]--;
    }

    for (i = 0; i < (int)this->size_; i++)
      this->seq_[i] = output[i];

    this->Write(count);
    count++;
  }
}
```

6.9 include/ShellSort.h File Reference

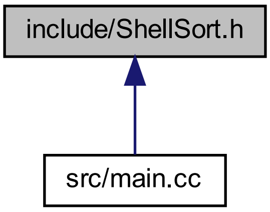
This file contains the ShellSort class.

```
#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

Fabrizzio Daniell Perilli Martin alu0101138589@ull.edu.es

Version

0.1

Date

2023-04-03

Copyright

6.10 ShellSort.h

Go to the documentation of this file.

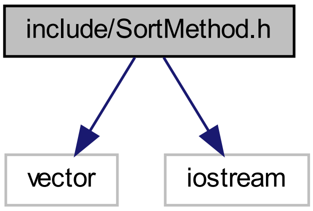
```

#include "SortMethod.h"

template <class Key>
class ShellSort : public SortMethod<Key>
{
public:
  ShellSort(std::vector<Key>, unsigned);
  void Sort() override;
};

template <class Key>
ShellSort<Key>::ShellSort(std::vector<Key> seq, unsigned size) : SortMethod<Key>(seq, size) {}


template <class Key>
void ShellSort<Key>::Sort()
{
  int count = 0;
  int alfa = this->size_ / 2;
  int aux, pos;
  std::cout « "Alfa: " « alfa « std::endl;
  while (alfa > 0)
  {
    for (size_t i = alfa; i < this->size_; i++)
    {
      pos = i;
      aux = this->seq_[i];
      while (pos >= alfa && aux < this->seq_[pos - alfa])
      {
        std::swap(this->seq_[pos], this->seq_[pos - alfa]);
        pos -= alfa;
      }
      this->Write(count);
      count++;
    }
    alfa /= 2;
    if (alfa > 0)
      std::cout « "Alfa: " « alfa « std::endl;
  }
}
```

6.11 include/SortMethod.h File Reference

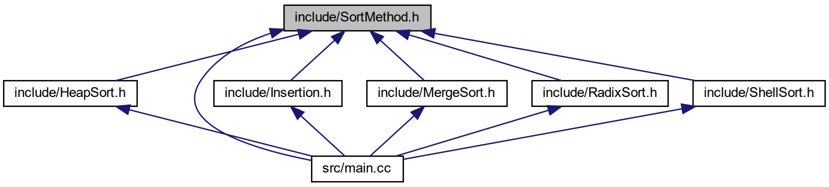
This a abstract class that contains the basic methods for the sorting algorithms.

```
#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 a abstract class that contains the basic methods for the sorting algorithms.

Author

Fabrizzio 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.

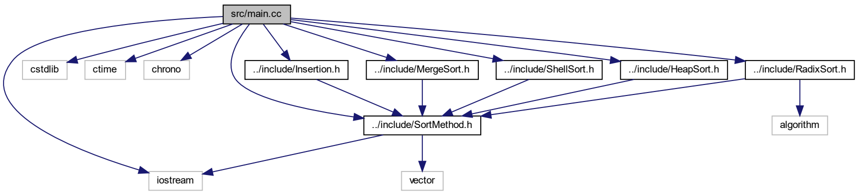
```

#include <vector>
#include <iostream>
#pragma once

template <class Key>
class SortMethod
{
protected:
  unsigned size_;
  std::vector<Key> seq_;

public:
  SortMethod() {}
  SortMethod(std::vector<Key>, unsigned);
  virtual void Sort() = 0;
  void Write(int);
};


template <class Key>
SortMethod<Key>::SortMethod(std::vector<Key> seq, unsigned size)
{
  seq_ = seq;
  size_ = size;
}


template <class Key>
void SortMethod<Key>::Write(int count)
{
  std::cout « "i: " « count « " -->";
  for (unsigned i = 0; i < size_; i++)
  {
    if (i == 0)
      std::cout « " | ";
    std::cout « seq_[i] « " | ";
  }
  std::cout « std::endl;
}
```

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

Fabrizzio Daniell Perilli Martin `alu0101138589@ull.edu.es`

Version

0.1

Date

2023-04-03

Copyright

Copyright (c) 2023

Index