Quadratic Sieve

Generated by Doxygen 1.8.6

Wed Mar 2 2016 02:39:12

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

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 files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 QS Namespace Reference

Namespaces

- numeric

Classes

- class QSFactor_base
- class QSAbstract_factor_base

Enumerations

- enum legendre_value { IS_NOT_QUADRATIC_RESIDUE }

5.1.1 Enumeration Type Documentation

5.1.1.1 enum QS::legendre_value

Enumerator

IS_NOT_QUADRATIC_RESIDUE

5.2 QS::numeric Namespace Reference

Classes

- class QSMatrix
- class QSVector
- class QSAbstract_matrix
- class QSAbstract_vector

Chapter 6

Class Documentation

6.1 Factor_base Class Reference

`#include <factor_base.h>`

6.1.1 Detailed Description

This class implements a factor base for algorighm of Quadratic Sieve. Factor base is a vector containing those primes number which have legendre number with N = 1, where N is the semiprime number we're going to factorize

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

- include/factor_base.h

6.2 QS::QSAbstract_factor_base Class Reference

`#include <abstract_factor_base.h>`

Public Member Functions

- QSAbstract_factor_base ()=delete
- QSAbstract_factor_base (mpz N)
- QSAbstract_factor_base (mpz_N, long unsigned upper_bound)
- unsigned long operator[] (unsigned int i) const

6.2.1 Constructor & Destructor Documentation

6.2.1.1 QS::QSAbstract_factor_base::QSAbstract_factor_base ( ) [delete]

6.2.1.2 QS::QSAbstract_factor_base::QSAbstract_factor_base (mpz *N*) [explicit]

6.2.1.3 QS::QSAbstract_factor_base::QSAbstract_factor_base (mpz_N , long unsigned *upper_bound*)

6.2.2 Member Function Documentation

6.2.2.1 unsigned long QS::QSAbstract_factor_base::operator[] (unsigned int *i*) const

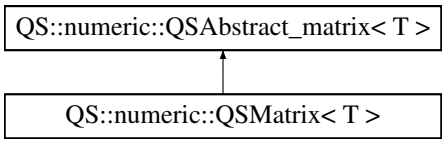
The documentation for this class was generated from the following file:

- include/virtual/abstract_factor_base.h

6.3 QS::numeric::QSAbstract_matrix< T > Class Template Reference

`#include <abstract_matrix.h>`

Inheritance diagram for QS::numeric::QSAbstract_matrix< T >:



Public Member Functions

- virtual QS::numeric::QSVector< T > sum_row (unsigned i, unsigned j) const =0

  sum two row src and dst and put result in dest
- virtual T get_elem (unsigned i, unsigned j)=0

  return element in position i,j
- virtual void put_elem (unsigned i, unsigned j, T elem)=0

  set the element in position i,j

Protected Attributes

- unsigned row_number

  number of rows
- unsigned col_number

  number of columns

6.3.1 Member Function Documentation

6.3.1.1 template<class T > virtual T QS::numeric::QSAbstract_matrix< T >::get_elem (unsigned *i*, unsigned *j*) [pure virtual]

return element in position i,j

Parameters

i	row indes
j	column index

Implemented in QS::numeric::QSMatrix< T >.

6.3.1.2 template<class T > virtual void QS::numeric::QSAbstract_matrix< T >::put_elem (unsigned *i*, unsigned *j*, T *elem*) [pure virtual]

set the element in position i,j

Parameters

i	row index
j	column index
elem	element to set

Implemented in QS::numeric::QSMatrix< T >.

6.3.1.3 template<class T > virtual QS::numeric::QSVector<T> QS::numeric::QSAbstract_matrix< T >::sum_row (unsigned *i*, unsigned *j*) const [pure virtual]

sum two row src and dst and put result in dest

Parameters

src	first row
dst	second row

Implemented in QS::numeric::QSMatrix< T >.

6.3.2 Member Data Documentation

6.3.2.1 template<class T > unsigned QS::numeric::QSAbstract_matrix< T >::col_number [protected]

number of columns

6.3.2.2 template<class T > unsigned QS::numeric::QSAbstract_matrix< T >::row_number [protected]

number of rows

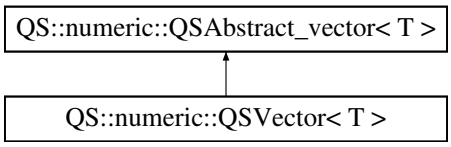
The documentation for this class was generated from the following file:

- include/virtual/abstract_matrix.h

6.4 QS::numeric::QSAbstract_vector< T > Class Template Reference

`#include <abstract_vector.h>`

Inheritance diagram for QS::numeric::QSAbstract_vector< T >:



Public Member Functions

- virtual void resize (unsigned dim)=0
- virtual unsigned size ()=0
- const T operator[] (unsigned i) const
- T & operator[] (unsigned i)

6.4.1 Member Function Documentation

6.4.1.1 template<class T > const T QS::numeric::QSAbstract_vector< T >::operator[] (unsigned *i*) const

6.4.1.2 template<class T > T& QS::numeric::QSAbstract_vector< T >::operator[] (unsigned *i*)

6.4.1.3 template<class T > virtual void QS::numeric::QSAbstract_vector< T >::resize (unsigned *dim*) [pure virtual]

Implemented in QS::numeric::QSVector< T >.

6.4.1.4 template<class T > virtual unsigned QS::numeric::QSAbstract_vector< T >::size ( ) [pure virtual]

Implemented in QS::numeric::QSVector< T >.

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

- include/virtual/abstract_vector.h

6.5 QS::QSFactor_base Class Reference

`#include <factor_base.h>`

Public Member Functions

- QSFactor_base ()=delete

  Factor_base constructor without elements has no sense. it needs at least a Number N.
- QSFactor_base (mpz N)
- QSFactor_base (mpz_N, long unsigned upper_bound)
- unsigned long operator[] (unsigned int i) const

6.5.1 Constructor & Destructor Documentation

6.5.1.1 QS::QSFactor_base::QSFactor_base ( ) [delete]

Factor_base constructor without elements has no sense. it needs at least a Number N.

6.5.1.2 QS::QSFactor_base::QSFactor_base (mpz *N*) [explicit]

6.5.1.3 QS::QSFactor_base::QSFactor_base (mpz_N , long unsigned *upper_bound*)

6.5.2 Member Function Documentation

6.5.2.1 unsigned long QS::QSFactor_base::operator[] (unsigned int *i*) const

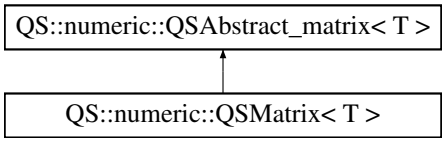
The documentation for this class was generated from the following file:

- include/factor_base.h

6.6 QS::numeric::QSMatrix< T > Class Template Reference

`#include <matrix.h>`

Inheritance diagram for QS::numeric::QSMatrix< T >:



Public Member Functions

- QSMatrix (unsigned i, unsigned j)
- QSMatrix (unsigned i, unsigned j, const T initial_value)
- T get_elem (unsigned i, unsigned j)

  return element in position i,j
- void put_elem (unsigned i, unsigned j, const T elem)

  set the element in position i,j
- unsigned get_row ()
- unsigned get_col ()
- QS::numeric::QSVector< T > sum_row (unsigned i, unsigned j) const

  sum two row src and dst and put result in dest

Additional Inherited Members

6.6.1 Constructor & Destructor Documentation

6.6.1.1 template<class T > QS::numeric::QSMatrix< T >::QSMatrix (unsigned *i*, unsigned *j*)

6.6.1.2 template<class T > QS::numeric::QSMatrix< T >::QSMatrix (unsigned *i*, unsigned *j*, const T *initial_value*)

6.6.2 Member Function Documentation

6.6.2.1 template<class T > unsigned QS::numeric::QSMatrix< T >::get_col ( ) [inline]

6.6.2.2 template<class T > T QS::numeric::QSMatrix< T >::get_elem (unsigned *i*, unsigned *j*) [virtual]

return element in position i,j

Parameters

i	row indes
j	column index

Implements QS::numeric::QSAbstract_matrix< T >.

6.6.2.3 template<class T > unsigned QS::numeric::QSMatrix< T >::get_row ( ) [inline]

6.6.2.4 template<class T > void QS::numeric::QSMatrix< T >::put_elem (unsigned *i*, unsigned *j*, const T *elem*) [virtual]

set the element in position i,j

Parameters

i	row index
j	column index
elem	element to set

Implements QS::numeric::QSAbstract_matrix< T >.

6.6.2.5 template<class T > QS::numeric::Vector< T > QS::numeric::QSMatrix< T >::sum_row (unsigned *i*, unsigned *j*) const [virtual]

sum two row src and dst and put result in dest

Parameters

src	first row
dst	second row

Implements QS::numeric::QSAbstract_matrix< T >.

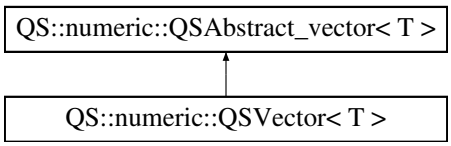
The documentation for this class was generated from the following file:

- include/matrix.h

6.7 QS::numeric::QSVector< T > Class Template Reference

`#include <vector.h>`

Inheritance diagram for QS::numeric::QSVector< T >:



Public Member Functions

- QSVector ()
- QSVector (unsigned dim)
- QSVector (unsigned dim, const T &initial_value)
- unsigned size ()
- void resize (unsigned s)
- void resize (unsigned s, T initial_value)
- T & get_elem (unsigned i)
- void set_elem (unsigned i, T elem)
- T & operator[] (unsigned i)
- const T & operator[] (unsigned i) const
- QSVector< T > operator+ (const QSVector< T > &v) const
- QSVector sum_row (QSVector< T > v1, QSVector< T > v2)
- void calc_wt ()
- void calc_lft_1_bit ()

6.7.1 Constructor & Destructor Documentation

6.7.1.1 template<class T> QS::numeric::QSVector< T >::QSVector ( )

6.7.1.2 template<class T > QS::numeric::QSVector< T >::QSVector (unsigned *dim*)

6.7.1.3 template<class T > QS::numeric::QSVector< T >::QSVector (unsigned *dim*, const T & *initial_value*)

6.7.2 Member Function Documentation

6.7.2.1 template<class T > void QS::numeric::QSVector< T >::calc_lft_1_bit ( )

6.7.2.2 template<class T > void QS::numeric::QSVector< T >::calc_wt ( )

6.7.2.3 template<class T > T & QS::numeric::QSVector< T >::get_elem (unsigned *i*)

6.7.2.4 template<class T> QSVector<T> QS::numeric::QSVector< T >::operator+ (const QSVector< T > & *v*) const

6.7.2.5 template<class T > T & QS::numeric::QSVector< T >::operator[] (unsigned *i*)

6.7.2.6 template<class T > const T & QS::numeric::QSVector< T >::operator[] (unsigned *i*) const

6.7.2.7 template<class T > void QS::numeric::QSVector< T >::resize (unsigned *s*) [virtual]

Implements QS::numeric::QSAbstract_vector< T >.

6.7.2.8 template<class T > void QS::numeric::QSVector< T >::resize (unsigned *s*, T *initial_value*)

6.7.2.9 template<class T > void QS::numeric::QSVector< T >::set_elem (unsigned *i*, T *elem*)

6.7.2.10 template<class T > unsigned QS::numeric::QSVector< T >::size ( ) [virtual]

Implements QS::numeric::QSAbstract_vector< T >.

6.7.2.11 template<class T> QSVector QS::numeric::QSVector< T >::sum_row (QSVector< T > *v1*, QSVector< T > *v2*)

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

- include/vector.h

Chapter 7

File Documentation

7.1 include/factor_base.h File Reference

```
#include <gmp.h>
#include "vector.h"
```

Classes

- class QS::QSFactor_base

Namespaces

- QS

Enumerations

- enum QS::legendre_value { QS::IS_NOT_QUADRATIC_RESIDUE }

7.2 include/matrix.h File Reference

```
#include "./virtual/abstract_matrix.h"
#include "./virtual/abstract_vector.h"
#include "template/matrix.templates.h"
```

Classes

- class QS::numeric::QSMatrix< T >

Namespaces

- QS
- QS::numeric

7.3 include/vector.h File Reference

```
#include "./virtual/abstract_vector.h"
#include <iostream>
#include <vector>
#include "template/vector.templates.h"
```

Classes

- class QS::numeric::QSVector< T >

Namespaces

- QS
- QS::numeric

7.4 include/virtual/abstract_factor_base.h File Reference

```
#include <gmp>
```

Classes

- class QS::QSAbstract_factor_base

Namespaces

- QS

7.5 include/virtual/abstract_matrix.h File Reference

Classes

- class QS::numeric::QSAbstract_matrix< T >

Namespaces

- QS
- QS::numeric

7.6 include/virtual/abstract_vector.h File Reference

Classes

- class QS::numeric::QSAbstract_vector< T >

Namespaces

- QS
- QS::numeric

7.7 src/factor_base.cpp File Reference

7.8 src/main.cpp File Reference

```
#include <iostream>
#include <gmp>
```

Functions

- int main ()

7.8.1 Function Documentation

7.8.1.1 int main ( )

7.9 src/matrix.cpp File Reference

7.10 src/test.cpp File Reference

```
#include <iostream>
#include "../include/vector.h"
#include "../include/matrix.h"
```

Functions

- int main ()

7.10.1 Function Documentation

7.10.1.1 int main ( )

Index