JSSP Solver

iota

Generated by Doxygen 1.8.14

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 Ui Namespace Reference

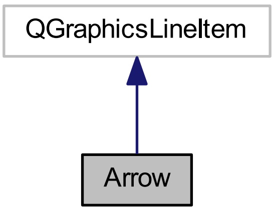
Chapter 6

Class Documentation

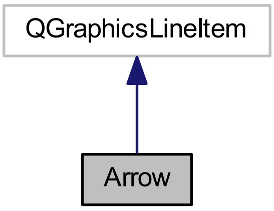
6.1 Arrow Class Reference

```
#include <arrow.h>
```

Inheritance diagram for Arrow:



Collaboration diagram for Arrow:



Public Types

- enum { Type = UserType + 1 }

Public Member Functions

- virtual int type () const
- Arrow (QGraphicsItem ∗parent=0)

  Creat an arrow.
- Arrow (const QPointF &startPoint, const QPointF &endPoint, QGraphicsItem ∗parent=0)

  Creat an arrow : The harder version.
- virtual QRectF boundingRect () const
- virtual QPainterPath shape () const
- QPointF startPoint () const
- void setStartPoint (const QPointF &startPoint)
- QPointF endPoint () const
- void setEndPoint (const QPointF &endPoint)

Protected Member Functions

- virtual void paint (QPainter ∗painter, const QStyleOptionGraphicsItem ∗option, QWidget ∗widget=0)

Private Attributes

- QPointF privateStartpoint
- QPointF privateEndpoint
- QPolygonF privateArrowhead

6.1.1 Detailed Description

Definition at line 22 of file arrow.h.

6.1.2 Member Enumeration Documentation

6.1.2.1 anonymous enum

`anonymous enum`

Enumerator

Type	

Definition at line 25 of file arrow.h.

6.1.3 Constructor & Destructor Documentation

6.1.3.1 Arrow() [1/2]

```
Arrow::Arrow (
            QGraphicsItem * parent = 0 )  [explicit]
```

Creat an arrow.

Parameters

parent	

Definition at line 22 of file arrow.cpp.

6.1.3.2 Arrow() [2/2]

```
Arrow::Arrow (
            const QPointF & startPoint,
            const QPointF & endPoint,
            QGraphicsItem * parent = 0 )
```

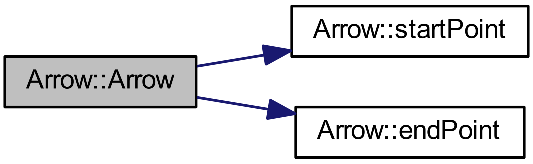
Creat an arrow : The harder version.

Parameters

startPoint	
endPoint	
parent	

Definition at line 34 of file arrow.cpp.

Here is the call graph for this function:



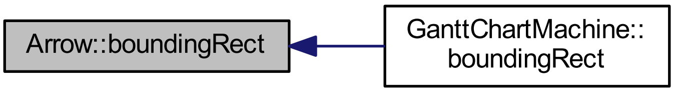
6.1.4 Member Function Documentation

6.1.4.1 boundingRect()

```
QRectF Arrow::boundingRect ( ) const  [virtual]
```

Definition at line 43 of file arrow.cpp.

Here is the caller graph for this function:

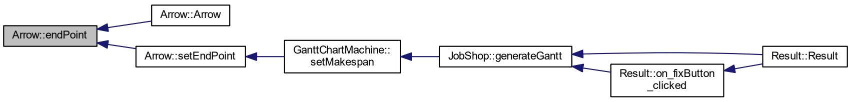


6.1.4.2 endPoint()

```
QPointF Arrow::endPoint ( ) const
```

Definition at line 88 of file arrow.cpp.

Here is the caller graph for this function:



6.1.4.3 paint()

```
void Arrow::paint (
            QPainter * painter,
            const QStyleOptionGraphicsItem * option,
            QWidget * widget = 0 )  [protected], [virtual]
```

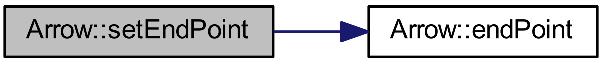
Definition at line 58 of file arrow.cpp.

6.1.4.4 setEndPoint()

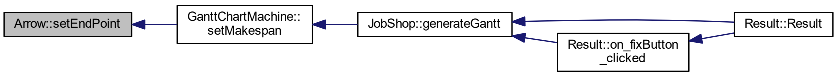
```
void Arrow::setEndPoint (
             const QPointF & endPoint )
```

Definition at line 92 of file arrow.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

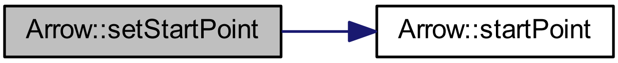


6.1.4.5 setStartPoint()

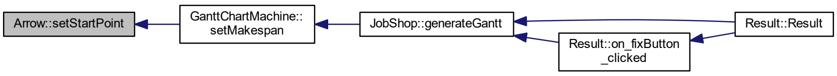
```
void Arrow::setStartPoint (
             const QPointF & startPoint )
```

Definition at line 100 of file arrow.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.4.6 shape()

```
QPainterPath Arrow::shape ( ) const  [virtual]
```

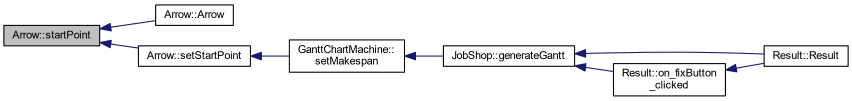
Definition at line 52 of file arrow.cpp.

6.1.4.7 startPoint()

```
QPointF Arrow::startPoint ( ) const
```

Definition at line 96 of file arrow.cpp.

Here is the caller graph for this function:



6.1.4.8 type()

```
virtual int Arrow::type ( ) const  [inline], [virtual]
```

Definition at line 26 of file arrow.h.

6.1.5 Member Data Documentation

6.1.5.1 privateArrowhead

```
QPolygonF Arrow::privateArrowhead  [private]
```

Definition at line 48 of file arrow.h.

6.1.5.2 privateEndpoint

```
QPointF Arrow::privateEndpoint  [private]
```

Definition at line 47 of file arrow.h.

6.1.5.3 privateStartpoint

```
QPointF Arrow::privateStartpoint  [private]
```

Definition at line 46 of file arrow.h.

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

- D:/Projects/JSSP-QT/src/include/arrow.h
- D:/Projects/JSSP-QT/src/gantt/arrow.cpp

6.2 BLIST Struct Reference

Public Attributes

- int machine
- int makespan
- int order [MAXJOB]

6.2.1 Detailed Description

Store the bottle informnation.

Definition at line 21 of file bottle.cpp.

6.2.2 Member Data Documentation

6.2.2.1 machine

```
int BLIST::machine
```

Machine number of this bottle.

Definition at line 22 of file bottle.cpp.

6.2.2.2 makespan

```
int BLIST::makespan
```

Makespan of this bottle.

Definition at line 23 of file bottle.cpp.

6.2.2.3 order

```
int BLIST::order[MAXJOB]
```

Job order of this bottle.

Definition at line 24 of file bottle.cpp.

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

- D:/Projects/JSSP-QT/src/algorithm/bottle.cpp

6.3 Fixer Struct Reference

```
#include <jobshop.h>
```

Public Attributes

- int machine
- int starttime
- int duration

6.3.1 Detailed Description

Definition at line 22 of file jobshop.h.

6.3.2 Member Data Documentation

6.3.2.1 duration

```
int Fixer::duration
```

Definition at line 25 of file jobshop.h.

6.3.2.2 machine

```
int Fixer::machine
```

Definition at line 23 of file jobshop.h.

6.3.2.3 starttime

```
int Fixer::starttime
```

Definition at line 24 of file jobshop.h.

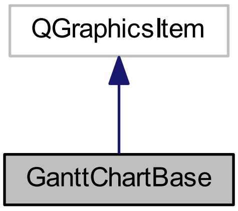
The documentation for this struct was generated from the following file:

- D:/Projects/JSSP-QT/src/include/jobshop.h

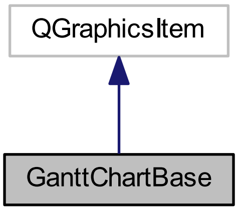
6.4 GanttChartBase Class Reference

```
#include <ganttchartbase.h>
```

Inheritance diagram for GanttChartBase:



Collaboration diagram for GanttChartBase:



Public Member Functions

- GanttChartBase (int makespan)

    Construct a gantt chart.
- virtual QRectF boundingRect () const

Static Public Member Functions

- static QPointF operationPosition (int time)
- static QPointF machineOffset ()

Static Public Attributes

- static const int widthUnit = 10
- static const int operationHeight = 20
- static const int machineHeight = operationHeight ∗ 3
- static const int machineHorizontalOffset = 35

Protected Member Functions

- virtual void paint (QPainter ∗painter, const QStyleOptionGraphicsItem ∗option, QWidget ∗widget=0)

  Paint this gantt chart.

Protected Attributes

- int makespan

6.4.1 Detailed Description

Definition at line 13 of file ganttchartbase.h.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 GanttChartBase()

```
GanttChartBase::GanttChartBase (
              int makespan )
```

Construct a gantt chart.

Parameters

makespan	The makespan of this project.

Definition at line 25 of file ganttchartbase.cpp.

6.4.3 Member Function Documentation

6.4.3.1 boundingRect()

```
QRectF GanttChartBase::boundingRect ( ) const  [virtual]
```

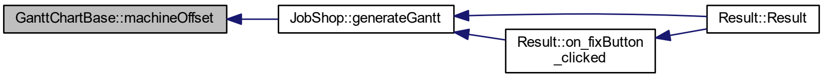
Definition at line 44 of file ganttchartbase.cpp.

6.4.3.2 machineOffset()

```
static QPointF GanttChartBase::machineOffset ( )  [inline], [static]
```

Definition at line 27 of file ganttchartbase.h.

Here is the caller graph for this function:

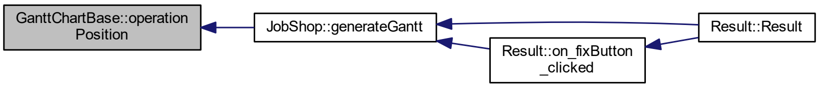


6.4.3.3 operationPosition()

```
static QPointF GanttChartBase::operationPosition (
            int time )  [inline], [static]
```

Definition at line 24 of file ganttchartbase.h.

Here is the caller graph for this function:


GanttChartBase::operation Position
JobShop::generateGantt
Result::on_fixButton _clicked
Result::Result

6.4.3.4 paint()

```
void GanttChartBase::paint (
            QPainter * painter,
            const QStyleOptionGraphicsItem * option,
            QWidget * widget = 0 )  [protected], [virtual]
```

Paint this gantt chart.

Parameters

painter	
option	
widget	

Definition at line 55 of file ganttchartbase.cpp.

6.4.4 Member Data Documentation

6.4.4.1 machineHeight

```
const int GanttChartBase::machineHeight = operationHeight * 3  [static]
```

Definition at line 21 of file ganttchartbase.h.

6.4.4.2 machineHorizontalOffset

```
const int GanttChartBase::machineHorizontalOffset = 35  [static]
```

Definition at line 22 of file ganttchartbase.h.

6.4.4.3 makespan

```
int GanttChartBase::makespan  [protected]
```

Definition at line 35 of file ganttchartbase.h.

6.4.4.4 operationHeight

```
const int GanttChartBase::operationHeight = 20  [static]
```

Definition at line 20 of file ganttchartbase.h.

6.4.4.5 widthUnit

```
const int GanttChartBase::widthUnit = 10  [static]
```

Definition at line 19 of file ganttchartbase.h.

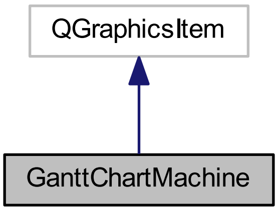
The documentation for this class was generated from the following files:

- D:/Projects/JSSP-QT/src/include/ganttchartbase.h
- D:/Projects/JSSP-QT/src/gantt/ganttchartbase.cpp

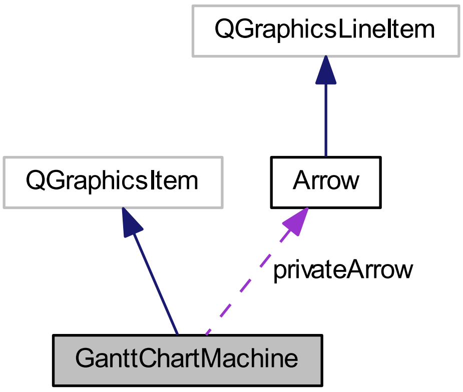
6.5 GanttChartMachine Class Reference

```
#include <ganttchartmachine.h>
```

Inheritance diagram for GanttChartMachine:



Collaboration diagram for GanttChartMachine:



Public Member Functions

- GanttChartMachine (const QString &id, QGraphicsItem ∗parent=0)

  Construct a machine class.
- ∼GanttChartMachine ()
- virtual QRectF boundingRect () const
- void setMakespan (int cMax)

  Set makespan of this machine.

Protected Member Functions

- virtual void paint (QPainter ∗painter, const QStyleOptionGraphicsItem ∗option, QWidget ∗widget=0)

  Paint this machine.

Private Attributes

- int makespan
- QString machineNum
- Arrow ∗ privateArrow

6.5.1 Detailed Description

Definition at line 15 of file ganttchartmachine.h.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 GanttChartMachine()

```
GanttChartMachine::GanttChartMachine (
             const QString & id,
             QGraphicsItem ∗ parent = 0 )
```

Construct a machine class.

Parameters

id	The given machine id.
parent	

Definition at line 23 of file ganttchartmachine.cpp.

6.5.2.2 ∼GanttChartMachine()

```
GanttChartMachine::∼GanttChartMachine ( )
```

Definition at line 41 of file ganttchartmachine.cpp.

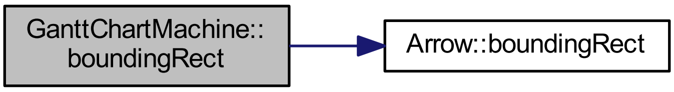
6.5.3 Member Function Documentation

6.5.3.1 boundingRect()

```
QRectF GanttChartMachine::boundingRect ( ) const  [virtual]
```

Definition at line 45 of file ganttchartmachine.cpp.

Here is the call graph for this function:



6.5.3.2 paint()

```
void GanttChartMachine::paint (
            QPainter * painter,
            const QStyleOptionGraphicsItem * option,
            QWidget * widget = 0 )  [protected], [virtual]
```

Paint this machine.

Parameters

painter	
option	
widget	

Definition at line 57 of file ganttchartmachine.cpp.

6.5.3.3 setMakespan()

```
void GanttChartMachine::setMakespan (
            int cMax )
```

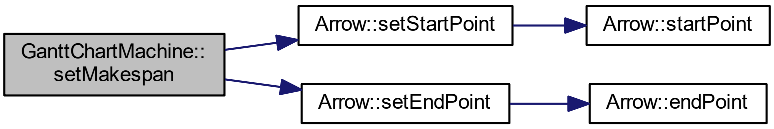
Set makespan of this machine.

Parameters

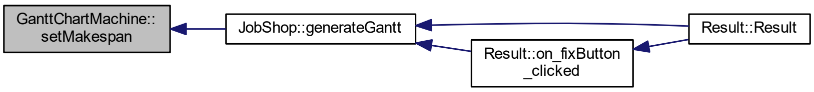
cMax	The makespan value.

Definition at line 68 of file ganttchartmachine.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.5.4 Member Data Documentation

6.5.4.1 machineNum

```
QString GanttChartMachine::machineNum  [private]
```

Definition at line 32 of file ganttchartmachine.h.

6.5.4.2 makespan

```
int GanttChartMachine::makespan  [private]
```

Definition at line 31 of file ganttchartmachine.h.

6.5.4.3 privateArrow

```
Arrow* GanttChartMachine::privateArrow  [private]
```

Definition at line 33 of file ganttchartmachine.h.

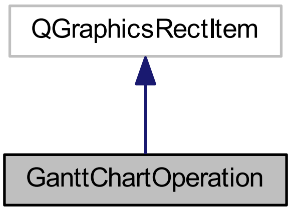
The documentation for this class was generated from the following files:

- D:/Projects/JSSP-QT/src/include/ganttchartmachine.h
- D:/Projects/JSSP-QT/src/gantt/ganttchartmachine.cpp

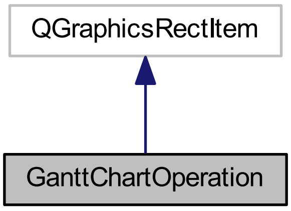
6.6 GanttChartOperation Class Reference

```
#include <ganttchartoperation.h>
```

Inheritance diagram for GanttChartOperation:



Collaboration diagram for GanttChartOperation:



Public Member Functions

- GanttChartOperation (const QString &id, int time, QColor color)

  Construct a GanttChartOperation class.

Protected Member Functions

- void paint (QPainter ∗painter, const QStyleOptionGraphicsItem ∗option, QWidget ∗widget)

  Paint this operation.

Protected Attributes

- QString privateId
- QColor privateColor

6.6.1 Detailed Description

Definition at line 14 of file ganttchartoperation.h.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 GanttChartOperation()

```
GanttChartOperation::GanttChartOperation (
             const QString & id,
             int time,
             QColor color )
```

Construct a GanttChartOperation class.

Parameters

id	Job id.
time	Start time.
color	Color of this operation.

Definition at line 21 of file ganttchartoperation.cpp.

6.6.3 Member Function Documentation

6.6.3.1 paint()

```
void GanttChartOperation::paint (
             QPainter * painter,
             const QStyleOptionGraphicsItem * option,
             QWidget * widget )  [protected]
```

Paint this operation.

Parameters

painter	
option	
widget	

Definition at line 35 of file ganttchartoperation.cpp.

6.6.4 Member Data Documentation

6.6.4.1 privateColor

```
QColor GanttChartOperation::privateColor  [protected]
```

Definition at line 25 of file ganttchartoperation.h.

6.6.4.2 privateId

```
QString GanttChartOperation::privateId  [protected]
```

Definition at line 24 of file ganttchartoperation.h.

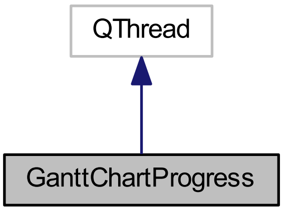
The documentation for this class was generated from the following files:

- D:/Projects/JSSP-QT/src/include/ganttchartoperation.h
- D:/Projects/JSSP-QT/src/gantt/ganttchartoperation.cpp

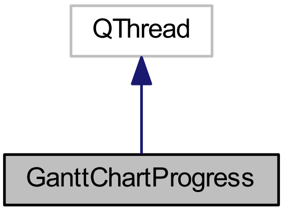
6.7 GanttChartProgress Class Reference

```
#include <ganttchartprogress.h>
```

Inheritance diagram for GanttChartProgress:



Collaboration diagram for GanttChartProgress:



Signals

- void updateline (int n)

Public Member Functions

- GanttChartProgress (double in_speed=1.0, QObject ∗parent=0)
- void setSpeed (double value)

  GanttChartProgress::setSpeed.

Protected Member Functions

- virtual void run () Q_DECL_OVERRIDE

  GanttChartProgress::run.

Private Attributes

- double speed

6.7.1 Detailed Description

Definition at line 14 of file ganttchartprogress.h.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 GanttChartProgress()

```
GanttChartProgress::GanttChartProgress (
             double in_speed = 1.0,
             QObject * parent = 0 )  [inline], [explicit]
```

Definition at line 17 of file ganttchartprogress.h.

6.7.3 Member Function Documentation

6.7.3.1 run()

```
void GanttChartProgress::run ( )  [protected], [virtual]
```

GanttChartProgress::run.

Definition at line 14 of file ganttchartprogress.cpp.

6.7.3.2 setSpeed()

```
void GanttChartProgress::setSpeed (
             double value )
```

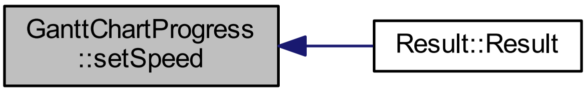
GanttChartProgress::setSpeed.

Parameters

value	

Definition at line 27 of file ganttchartprogress.cpp.

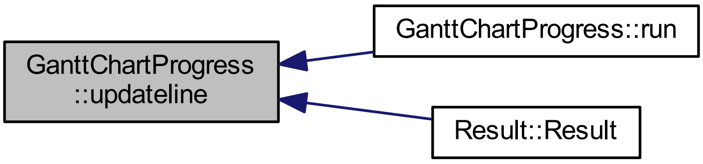
Here is the caller graph for this function:



6.7.3.3 updateline

```
void GanttChartProgress::updateline (
             int n )  [signal]
```

Here is the caller graph for this function:



6.7.4 Member Data Documentation

6.7.4.1 speed

```
double GanttChartProgress::speed  [private]
```

Definition at line 25 of file ganttchartprogress.h.

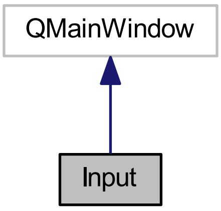
The documentation for this class was generated from the following files:

- D:/Projects/JSSP-QT/src/include/ganttchartprogress.h
- D:/Projects/JSSP-QT/src/gantt/ganttchartprogress.cpp

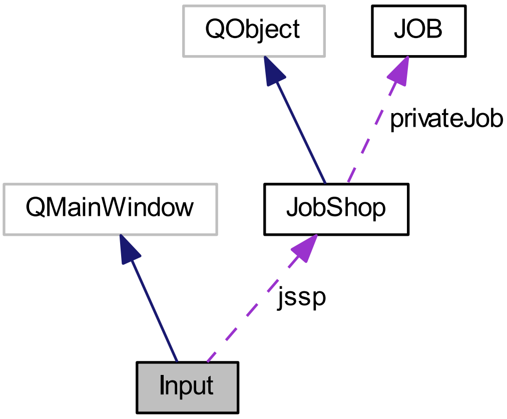
6.8 Input Class Reference

```
#include <input.h>
```

Inheritance diagram for Input:



Collaboration diagram for Input:



Public Member Functions

- Input (QWidget ∗parent=0)

  Set the input window.
- ∼Input ()

  Delete the window.

Private Slots

- void on_start_button_clicked ()

  Action of start button.

Private Attributes

- Ui::Input ∗ ui
- JobShop ∗ jssp

6.8.1 Detailed Description

Definition at line 19 of file input.h.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 Input()

```
Input::Input (
            QWidget ∗ parent = 0 )  [explicit]
```

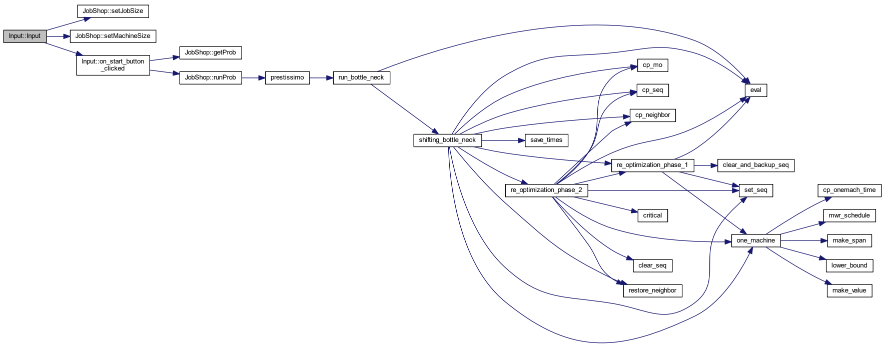
Set the input window.

Parameters

parent	

Definition at line 18 of file input.cpp.

Here is the call graph for this function:



6.8.2.2 ∼Input()

```
Input::∼Input ( )
```

Delete the window.

Definition at line 32 of file input.cpp.

6.8.3 Member Function Documentation

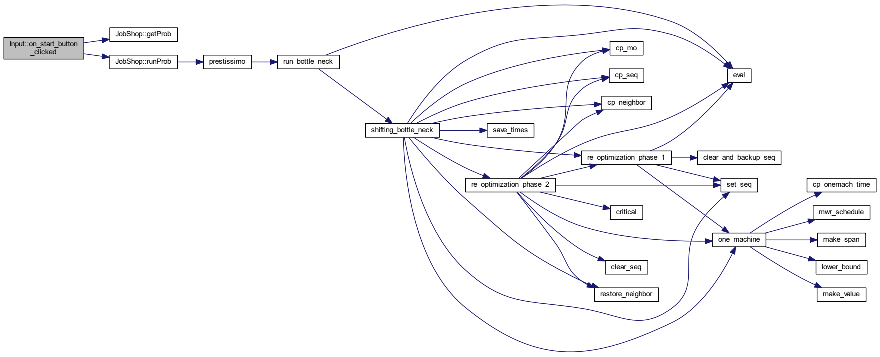
6.8.3.1 on_start_button_clicked

```
void Input::on_start_button_clicked ( )  [private], [slot]
```

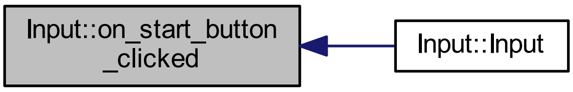
Action of start button.

Definition at line 41 of file input.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.8.4 Member Data Documentation

6.8.4.1 jssp

```
JobShop* Input::jssp  [private]
```

Definition at line 32 of file input.h.

6.8.4.2 ui

```
Ui::Input* Input::ui  [private]
```

Definition at line 31 of file input.h.

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

- D:/Projects/JSSP-QT/src/include/input.h
- D:/Projects/JSSP-QT/src/ui/input.cpp

6.9 JOB Struct Reference

```
#include <bottle.h>
```

Public Attributes

- int estime [MAXMACHINE]
- int mhtime [MAXMACHINE]
- int magic [MAXMACHINE]
- int order [MAXMACHINE]
- int process_time [MAXMACHINE]
- int step [MAXMACHINE]
- int next [MAXMACHINE]
- int prev [MAXMACHINE]
- int start [MAXMACHINE]

6.9.1 Detailed Description

Data representation for a job.

Definition at line 41 of file bottle.h.

6.9.2 Member Data Documentation

6.9.2.1 estime

```
int JOB::estime[MAXMACHINE]
```

Earlist starting time of this job on each machine. Which is simply the sum of this job's processing times on the machine before [order[machine]] in this jobs prescribed ordering.

Definition at line 42 of file bottle.h.

6.9.2.2 magic

```
int JOB::magic[MAXMACHINE]
```

The number generated and managed by the God in the computer. Every one who changed the name of this feild will be seen as an evil and will be cursed by the God.

Definition at line 44 of file bottle.h.

6.9.2.3 mhtime

```
int JOB::mhtime[MAXMACHINE]
```

Minimum halting time of this job after [machine num]. Which is simply the sum of this job's processing times on the machine after [order[machine]] in this jobs prescribed ordering.

Definition at line 43 of file bottle.h.

6.9.2.4 next

```
int JOB::next[MAXMACHINE]
```

Next job on machine [i].

Definition at line 48 of file bottle.h.

6.9.2.5 order

```
int JOB::order[MAXMACHINE]
```

Required machine order for the job.

Definition at line 45 of file bottle.h.

6.9.2.6 prev

```
int JOB::prev[MAXMACHINE]
```

Previous job blah blah.

Definition at line 49 of file bottle.h.

6.9.2.7 process_time

```
int JOB::process_time[MAXMACHINE]
```

Process time of each machine.

Definition at line 46 of file bottle.h.

6.9.2.8 start

```
int JOB::start[MAXMACHINE]
```

Start time of this job on each machine.

Definition at line 50 of file bottle.h.

6.9.2.9 step

```
int JOB::step[MAXMACHINE]
```

Solution step indexed by machine.

Definition at line 47 of file bottle.h.

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

- D:/Projects/JSSP-QT/src/include/bottle.h

6.10 JOBMACHINEPAR Struct Reference

Public Attributes

- int job
- int machine

6.10.1 Detailed Description

Auxiliary struct to calcuate the makespan.

Definition at line 13 of file eval.cpp.

6.10.2 Member Data Documentation

6.10.2.1 job

```
int JOBMACHINEPAR::job
```

Definition at line 14 of file eval.cpp.

6.10.2.2 machine

```
int JOBMACHINEPAR::machine
```

Definition at line 15 of file eval.cpp.

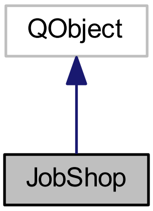
The documentation for this struct was generated from the following file:

- D:/Projects/JSSP-QT/src/algorithm/eval.cpp

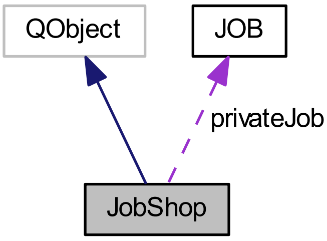
6.11 JobShop Class Reference

```
#include <jobshop.h>
```

Inheritance diagram for JobShop:



Collaboration diagram for JobShop:



Public Member Functions

- JobShop ()

  Get pointers of previous defined varibles.
- job_t ∗ getJob ()
- int getJobSize ()
- int getMachineSize ()
- void setJobSize (int i)

  Set job size using the given value.
- void setMachineSize (int i)

  Set machine size using the given value.
- void getProb (QString)

  Read innstance file from the given str.
- GanttChartBase ∗ generateGantt ()

  Generate the gantt chart from our result.
- void runProb ()

  Run the awesome soler algorithm.
- QString colsoleOutput ()
- void writeFile (QString file_name)

Public Attributes

- QVector< Fixer ∗ > fixer

Private Attributes

- job_t ∗ privateJob
- int ∗ privateJobSize
- int ∗ privateMachineSize

6.11.1 Detailed Description

Definition at line 29 of file jobshop.h.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 JobShop()

```
JobShop::JobShop ( )
```

Get pointers of previous defined varibles.

Definition at line 23 of file jobshop.cpp.

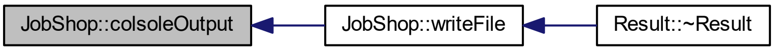
6.11.3 Member Function Documentation

6.11.3.1 colsoleOutput()

```
QString JobShop::colsoleOutput ( )
```

Definition at line 77 of file io.cpp.

Here is the caller graph for this function:



6.11.3.2 generateGantt()

```
GanttChartBase * JobShop::generateGantt ( )
```

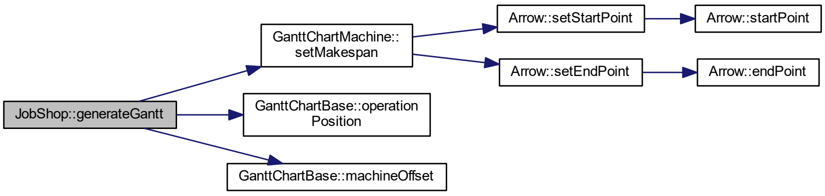
Generate the gantt chart from our result.

Returns

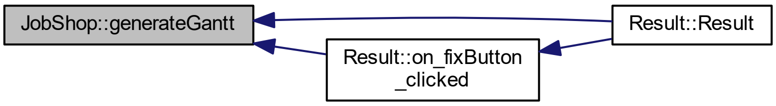
The gantt chart.

Definition at line 76 of file jobshop.cpp.

Here is the call graph for this function:


JobShop::generateGantt
GanttChartMachine::setMakespan
Arrow::setStartPoint
Arrow::startPoint
Arrow::setEndPoint
Arrow::endPoint
GanttChartBase::operationPosition
GanttChartBase::machineOffset

Here is the caller graph for this function:



6.11.3.3 getJob()

```
job_t* JobShop::getJob ( )  [inline]
```

Definition at line 34 of file jobshop.h.

6.11.3.4 getJobSize()

```
int JobShop::getJobSize ( )  [inline]
```

Definition at line 35 of file jobshop.h.

6.11.3.5 getMachineSize()

```
int JobShop::getMachineSize ( )  [inline]
```

Definition at line 36 of file jobshop.h.

6.11.3.6 getProb()

```
void JobShop::getProb (
             QString str )
```

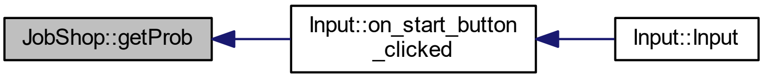
Read innstance file from the given str.

Parameters

str	The given string grom the QPlainTextEdit

Definition at line 51 of file jobshop.cpp.

Here is the caller graph for this function:



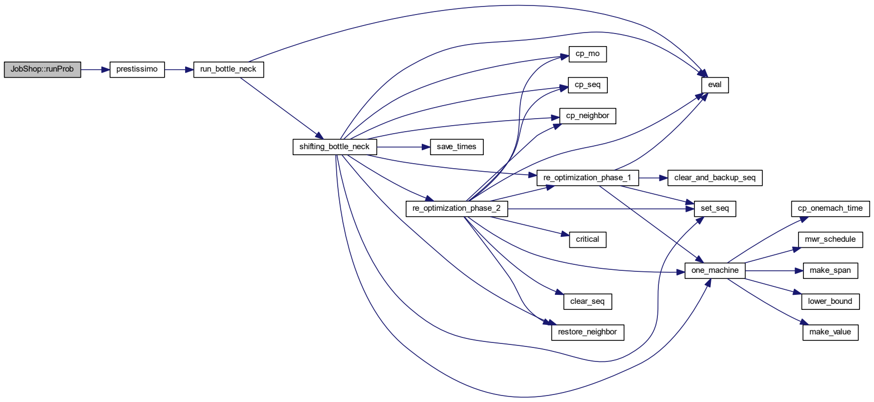
6.11.3.7 runProb()

```
void JobShop::runProb ( )
```

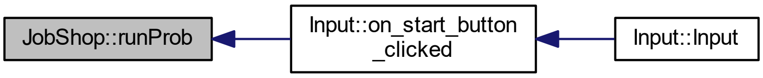
Run the awesome soler algorithm.

Definition at line 68 of file jobshop.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.11.3.8 setJobSize()

```
void JobShop::setJobSize (
              int i )
```

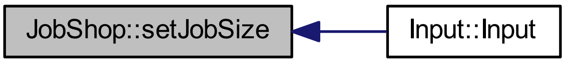
Set job size using the given value.

Parameters

i	Given value from the spin box

Definition at line 35 of file jobshop.cpp.

Here is the caller graph for this function:



6.11.3.9 setMachineSize()

```
void JobShop::setMachineSize (
             int i )
```

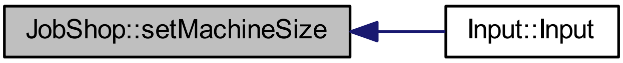
Set machine size using the given value.

Parameters

i	Given value from the spin box

Definition at line 43 of file jobshop.cpp.

Here is the caller graph for this function:



6.11.3.10 writeFile()

```
void JobShop::writeFile (
             QString file_name )
```

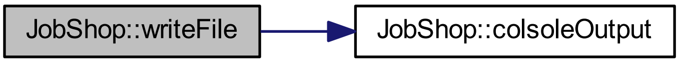
Print result to file...

Parameters

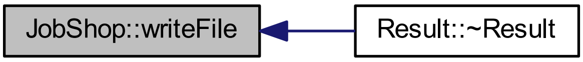
file_name	Instance file path
pairs	Pair to be printed...

Definition at line 134 of file io.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



6.11.4 Member Data Documentation

6.11.4.1 fixer

```
QVector<Fixer*> JobShop::fixer
```

Definition at line 44 of file jobshop.h.

6.11.4.2 privateJob

```
job_t* JobShop::privateJob  [private]
```

Definition at line 46 of file jobshop.h.

6.11.4.3 privateJobSize

```
int* JobShop::privateJobSize  [private]
```

Definition at line 47 of file jobshop.h.

6.11.4.4 privateMachineSize

```
int* JobShop::privateMachineSize [private]
```

Definition at line 48 of file jobshop.h.

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

- D:/Projects/JSSP-QT/src/include/jobshop.h
- D:/Projects/JSSP-QT/src/console/io.cpp
- D:/Projects/JSSP-QT/src/wrapper/jobshop.cpp

6.12 MACHINEORDER Struct Reference

Public Attributes

- int size
- int machines [MAXMACHINE]

6.12.1 Detailed Description

Machine order type.

Definition at line 30 of file bottle.cpp.

6.12.2 Member Data Documentation

6.12.2.1 machines

```
int MACHINEORDER::machines[MAXMACHINE]
```

Sequenced machine list.

Definition at line 32 of file bottle.cpp.

6.12.2.2 size

```
int MACHINEORDER::size
```

Sequenced machine number.

Definition at line 31 of file bottle.cpp.

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

- D:/Projects/JSSP-QT/src/algorithm/bottle.cpp

6.13 NEIGHBOR Struct Reference

Public Attributes

- int next [MAXMACHINE]
- int prev [MAXMACHINE]

6.13.1 Detailed Description

A temporary struct to store a sequence.

Definition at line 39 of file bottle.cpp.

6.13.2 Member Data Documentation

6.13.2.1 next

```
int NEIGHBOR::next[MAXMACHINE]
```

You will be either silly or able to understand the name's mean.

Definition at line 40 of file bottle.cpp.

6.13.2.2 prev

```
int NEIGHBOR::prev[MAXMACHINE]
```

Same as the previous one.

Definition at line 41 of file bottle.cpp.

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

- D:/Projects/JSSP-QT/src/algorithm/bottle.cpp

6.14 ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT Struct Reference

Public Attributes

- int active
- int bound

6.14.1 Detailed Description

Info of node of a branch and bound tree.

Definition at line 22 of file onemachine.cpp.

6.14.2 Member Data Documentation

6.14.2.1 active

```
int ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT::active
```

Wether this node is active

Definition at line 23 of file onemachine.cpp.

6.14.2.2 bound

```
int ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT::bound
```

See "JOBTYPE" for more info.

Definition at line 24 of file onemachine.cpp.

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

- D:/Projects/JSSP-QT/src/algorithm/onemachine.cpp

6.15 ONEMACHINEstime Struct Reference

```
#include <bottle.h>
```

Public Attributes

- int estime [MAXJOB]
- int mhtime [MAXJOB]
- int process_time [MAXJOB]

6.15.1 Detailed Description

Store the time info for every job runs on the same machine.

Definition at line 63 of file bottle.h.

6.15.2 Member Data Documentation

6.15.2.1 estime

```
int ONEMACHINEstime::estime[MAXJOB]
```

See "JOBTYPE" for more info.

Definition at line 64 of file bottle.h.

6.15.2.2 mhtime

```
int ONEMACHINEstime::mhtime[MAXJOB]
```

See "JOBTYPE" for more info.

Definition at line 65 of file bottle.h.

6.15.2.3 process_time

```
int ONEMACHINEstime::process_time[MAXJOB]
```

See "JOBTYPE" for more info.

Definition at line 66 of file bottle.h.

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

- D:/Projects/JSSP-QT/src/include/bottle.h

6.16 pair Struct Reference

The pair struct for sorting machines.

Public Attributes

- int starttime
- int endtime

6.16.1 Detailed Description

The pair struct for sorting machines.

Definition at line 26 of file result.cpp.

6.16.2 Member Data Documentation

6.16.2.1 endtime

```
int pair::endtime
```

Definition at line 29 of file result.cpp.

6.16.2.2 starttime

```
int pair::starttime
```

Definition at line 28 of file result.cpp.

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

- D:/Projects/JSSP-QT/src/ui/result.cpp

6.17 PAIR_ASSISTANT_TYPE Struct Reference

Public Attributes

- int start_time
- int job_num
- int mach_num
- int proc_time
- int step

6.17.1 Detailed Description

A temporary struct for converting the internal representation of the solution to the format required by those soreheads.

Definition at line 24 of file io.cpp.

6.17.2 Member Data Documentation

6.17.2.1 job_num

```
int PAIR_ASSISTANT_TYPE::job_num
```

Serial number of the job of this node.

Definition at line 26 of file io.cpp.

6.17.2.2 mach_num

```
int PAIR_ASSISTANT_TYPE::mach_num
```

Serial number of the machine of this node.

Definition at line 27 of file io.cpp.

6.17.2.3 proc_time

```
int PAIR_ASSISTANT_TYPE::proc_time
```

Process time (a.k.a duration) of this node.

Definition at line 28 of file io.cpp.

6.17.2.4 start_time

```
int PAIR_ASSISTANT_TYPE::start_time
```

Start time of this node.

Definition at line 25 of file io.cpp.

6.17.2.5 step

```
int PAIR_ASSISTANT_TYPE::step
```

Serial number of the order of this node in the job.

Definition at line 29 of file io.cpp.

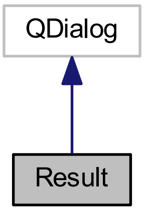
The documentation for this struct was generated from the following file:

- D:/Projects/JSSP-QT/src/console/io.cpp

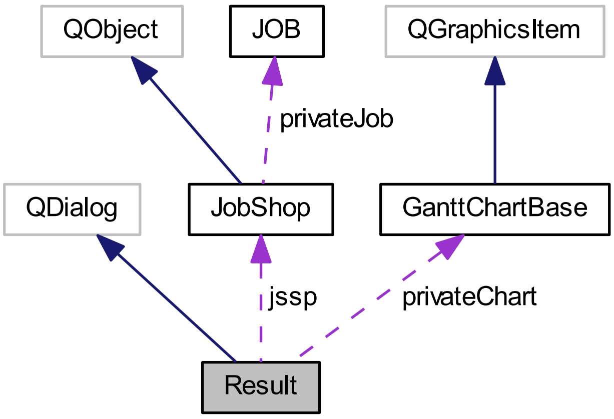
6.18 Result Class Reference

```
#include <result.h>
```

Inheritance diagram for Result:



Collaboration diagram for Result:



Public Member Functions

- Result (JobShop ∗instance, QWidget ∗parent=0)

    Construct the result window.
- void Fix (int machine, int clock, int duration)

    Handle the fix command.
- ∼Result ()

    Delete the window.

Private Slots

- void on_fixButton_clicked ()

    The action of fixButton.
- void on_line_updated (int time)

Private Attributes

- Ui::Result ∗ ui
- JobShop ∗ jssp
- GanttChartBase ∗ privateChart
- QGraphicsLineItem ∗ privateLine
- QGraphicsScene ∗ privateScene

6.18.1 Detailed Description

Definition at line 26 of file result.h.

6.18.2 Constructor & Destructor Documentation

6.18.2.1 Result()

```
Result::Result (
            JobShop ∗ instance,
            QWidget ∗ parent = 0 )
```

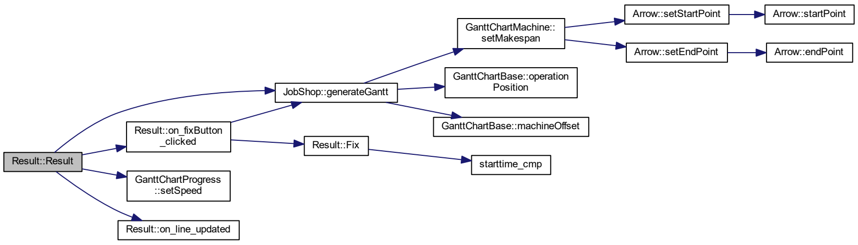
Construct the result window.

Parameters

instance	The instance class.
parent	...

Definition at line 48 of file result.cpp.

Here is the call graph for this function:



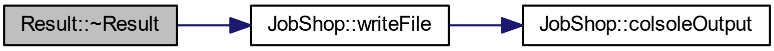
6.18.2.2 ∼Result()

```
Result::∼Result ( )
```

Delete the window.

Definition at line 77 of file result.cpp.

Here is the call graph for this function:



6.18.3 Member Function Documentation

6.18.3.1 Fix()

```
void Result::Fix (
             int machine,
             int clock,
             int duration )
```

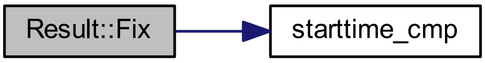
Handle the fix command.

Parameters

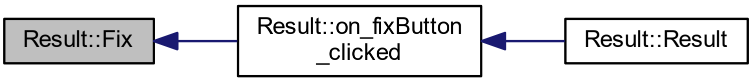
machine	
clock	
duration	

Definition at line 90 of file result.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



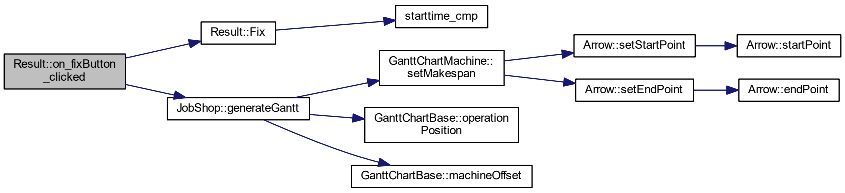
6.18.3.2 on_fixButton_clicked

```
void Result::on_fixButton_clicked ( )  [private], [slot]
```

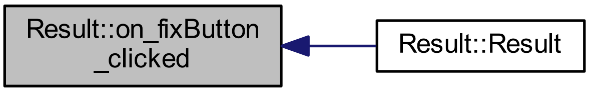
The action of fixButton.

Definition at line 156 of file result.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:

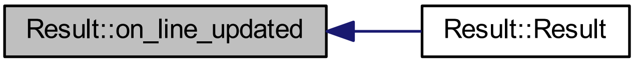


6.18.3.3 on_line_updated

```
void Result::on_line_updated (
             int time )  [private], [slot]
```

Definition at line 171 of file result.cpp.

Here is the caller graph for this function:



6.18.4 Member Data Documentation

6.18.4.1 jssp

```
JobShop* Result::jssp  [private]
```

Definition at line 41 of file result.h.

6.18.4.2 privateChart

```
GanttChartBase* Result::privateChart  [private]
```

Definition at line 42 of file result.h.

6.18.4.3 privateLine

```
QGraphicsLineItem* Result::privateLine  [private]
```

Definition at line 43 of file result.h.

6.18.4.4 privateScene

```
QGraphicsScene* Result::privateScene  [private]
```

Definition at line 44 of file result.h.

6.18.4.5 ui

```
Ui::Result* Result::ui  [private]
```

Definition at line 40 of file result.h.

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

- D:/Projects/JSSP-QT/src/include/result.h
- D:/Projects/JSSP-QT/src/ui/result.cpp

6.19 SEQUENCE Struct Reference

```
#include <bottle.h>
```

Public Attributes

- int job [MAXJOB]

6.19.1 Detailed Description

Job sequences on a machine.

Definition at line 56 of file bottle.h.

6.19.2 Member Data Documentation

6.19.2.1 job

```
int SEQUENCE::job[MAXJOB]
```

Job sequences on a machine.

Definition at line 57 of file bottle.h.

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

- D:/Projects/JSSP-QT/src/include/bottle.h

Chapter 7

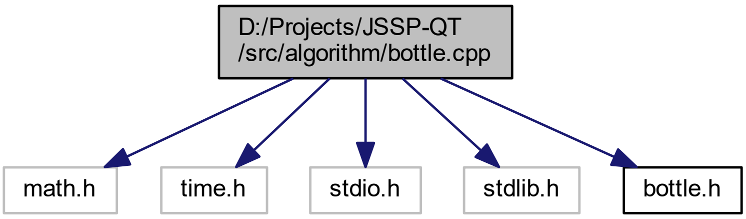
File Documentation

7.1 D:/Projects/JSSP-QT/src/algorithm/bottle.cpp File Reference

Core algorithms.

```
#include <math.h>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <bottle.h>
```

Include dependency graph for bottle.cpp:



Classes

- struct BLIST
- struct MACHINEORDER
- struct NEIGHBOR

Macros

- #define TRY_COUNT 10

Typedefs

- typedef struct BLIST blist_t
- typedef struct MACHINEORDER mo_t
- typedef struct NEIGHBOR neighbor_t

Functions

- static void shifting_bottle_neck (sequence_t ∗seq, mo_t ∗machine_order, int ∗try_time_set)

  The major implementation of the Shifting Bottleneck Procedure, with a backtracing method.
- static void clear_and_backup_seq (sequence_t ∗seq, int mach, int ∗save)
- static void clear_seq (sequence_t ∗seq, int machine)
- static void cp_mo (mo_t ∗mew, mo_t ∗origin)
- static void cp_seq (sequence_t ∗mew, sequence_t ∗origin)
- static void cp_neighbor (neighbor_t ∗mew)
- static void re_optimization_phase_1 (sequence_t ∗seq, mo_t ∗mo, int ∗makespan)
- static void set_seq (sequence_t ∗seq, int mach, int ∗order)
- static void save_times (void)
- static void re_optimization_phase_2 (sequence_t ∗seq, mo_t ∗mo, int ∗makespan)
- static void restore_neighbor (neighbor_t ∗old)
- static int critical (int machine, int makespan)
- int eval (sequence_t ∗seq)

Variables

- int best_makespan = INFINITAS

  Store the best makespan value.

7.1.1 Detailed Description

Core algorithms.

Core algorithms to solve the JSSP.

Author

Name1e5s

7.1.2 Macro Definition Documentation

7.1.2.1 TRY_COUNT

```
#define TRY_COUNT 10
```

Definition at line 16 of file bottle.cpp.

7.1.3 Typedef Documentation

7.1.3.1 blist_t

```
typedef struct BLIST blist_t
```

Store the bottle informnation.

7.1.3.2 mo_t

```
typedef struct MACHINEORDER mo_t
```

Machine order type.

7.1.3.3 neighbor_t

```
typedef struct NEIGHBOR neighbor_t
```

A temporary struct to store a sequence.

7.1.4 Function Documentation

7.1.4.1 clear_and_backup_seq()

```
static void clear_and_backup_seq (
             sequence_t * seq,
             int machine,
             int * save )  [inline], [static]
```

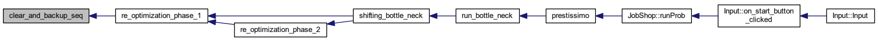
Store current sequence of machine N in the given address. Then just clear the sequence.

Parameters

seq	Sequence to be cleared.
machine	Current machine number.
save	Address to save the old sequence. A NULL address means the old sequence won't be stored.

Definition at line 368 of file bottle.cpp.

Here is the caller graph for this function:



7.1.4.2 clear_seq()

```
static void clear_seq (
             sequence_t * seq,
             int machine )  [inline], [static]
```

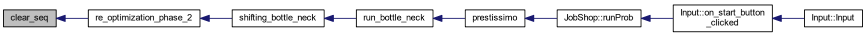
Clear the sequence.

Parameters

seq	Sequence to be cleared.
machine	Current machine number.

Definition at line 384 of file bottle.cpp.

Here is the caller graph for this function:



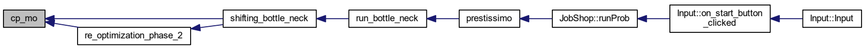
7.1.4.3 cp_mo()

```
static void cp_mo (
             mo_t * mew,
             mo_t * origin )  [inline], [static]
```

Copy the origin machine order to mew.

Definition at line 354 of file bottle.cpp.

Here is the caller graph for this function:



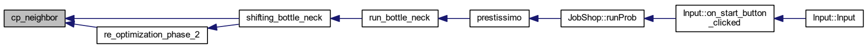
7.1.4.4 cp_neighbor()

```
static void cp_neighbor (
            neighbor_t * mew )  [inline], [static]
```

Store neighbor to a neighbor_t varible.

Definition at line 423 of file bottle.cpp.

Here is the caller graph for this function:



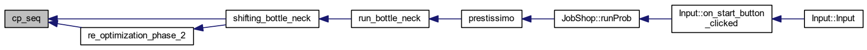
7.1.4.5 cp_seq()

```
static void cp_seq (
            sequence_t * mew,
            sequence_t * origin )  [inline], [static]
```

Copy the origin sequence order to mew.

Definition at line 396 of file bottle.cpp.

Here is the caller graph for this function:



7.1.4.6 critical()

```
static int critical (
            int machine,
            int makespan )  [inline], [static]
```

Test wether the machine is the critical machine, which means the end of the procedure of this machine is also the end of all the operations.

Parameters

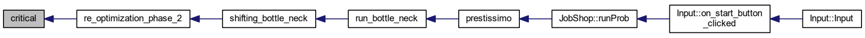
machine	Machine number to be tested.
makespan	The given makespan

Returns

If the machine is the critical machine, return 1. Else return 0.

Definition at line 344 of file bottle.cpp.

Here is the caller graph for this function:



7.1.4.7 eval()

```
int eval (
             sequence_t * seq )
```

Evaluate the makespan of the given sequence.

Parameters

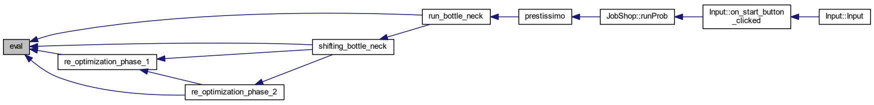
seq	The sequence of job.

Returns

The makespan of the sequence.

Definition at line 31 of file eval.cpp.

Here is the caller graph for this function:



7.1.4.8 re_optimization_phase_1()

```
static void re_optimization_phase_1 (
             sequence_t * seq,
             mo_t * machine_order,
             int * makespan )  [inline], [static]
```

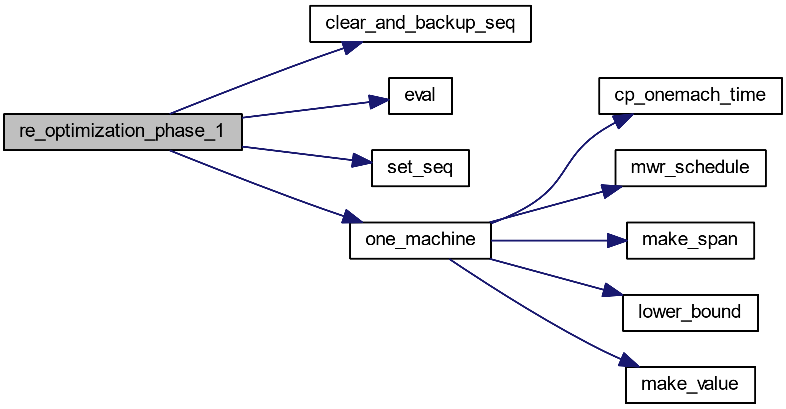
The re-optimization... Phase 1

Parameters

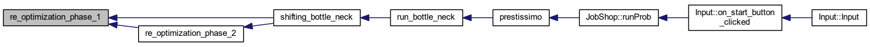
seq	The sequence
machine_order	Machine order
makespan	Current makespan

Definition at line 219 of file bottle.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.4.9 re_optimization_phase_2()

```
static void re_optimization_phase_2 (
            sequence_t * seq,
            mo_t * machine_order,
            int * makespan )  [inline], [static]
```

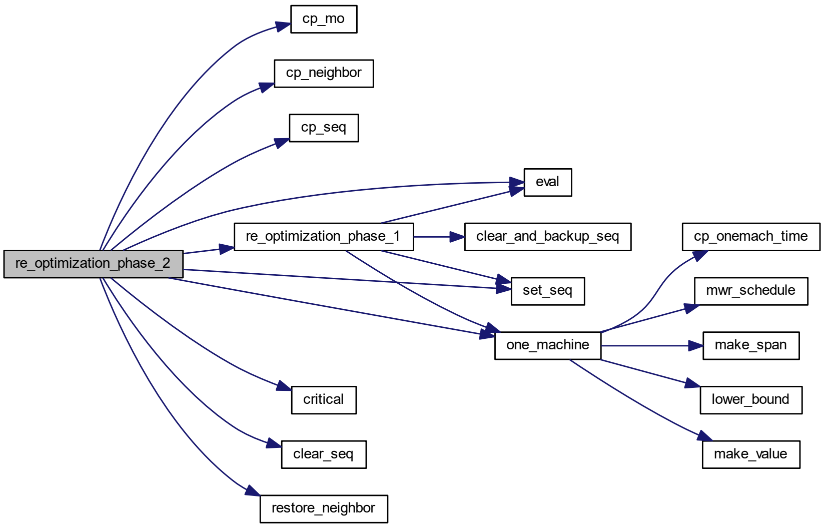
The re-optimization... Phase 2

Parameters

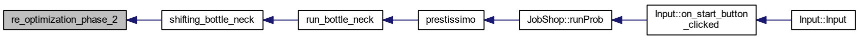
seq	The sequence
machine_order	Machine order
makespan	Current makespan

Definition at line 284 of file bottle.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



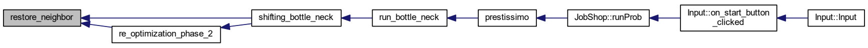
7.1.4.10 restore_neighbor()

```
static void restore_neighbor (
            neighbor_t * old )  [inline], [static]
```

Load neighbor from a neighbor_t varible.

Definition at line 434 of file bottle.cpp.

Here is the caller graph for this function:



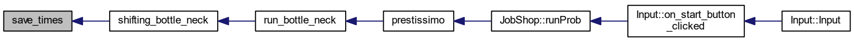
7.1.4.11 save_times()

```
static void save_times (
             void   )  [inline], [static]
```

Save current start time of each operation.

Definition at line 446 of file bottle.cpp.

Here is the caller graph for this function:



7.1.4.12 set_seq()

```
static void set_seq (
             sequence_t * seq,
             int machine,
             int * order )  [inline], [static]
```

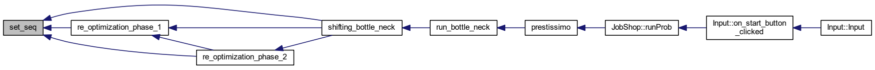
Set sequence by the given order.

Parameters

seq	Sequence to be set.
machine	The machine which the sequence relies on.
order	The given order.

Definition at line 409 of file bottle.cpp.

Here is the caller graph for this function:



7.1.4.13 shifting_bottle_neck()

```
static void shifting_bottle_neck (
             sequence_t * seq,
             mo_t * machine_order,
             int * try_time_set )  [inline], [static]
```

The major implementation of the Shifting Bottleneck Procedure, with a backtracing method.

The basic idea of the algorithm can be described as follows: It sequences the machines one by one successively, taking each time the machine identified as a bottleneck among the machine not yet sequenced. Every time after a mew machine is sequenced, all previously sequenced sequence will be locally re-optimized. Bottleneck identification and the local re-optimization are both based on solving a one machine scheduling problem, which is more easy than the JSSP. In this implementation a backtracing trick is introduced to improve the quality of the solution, which give us a method to use a slightly more time to run the basic shifting bottleneck procedure more times.

Parameters

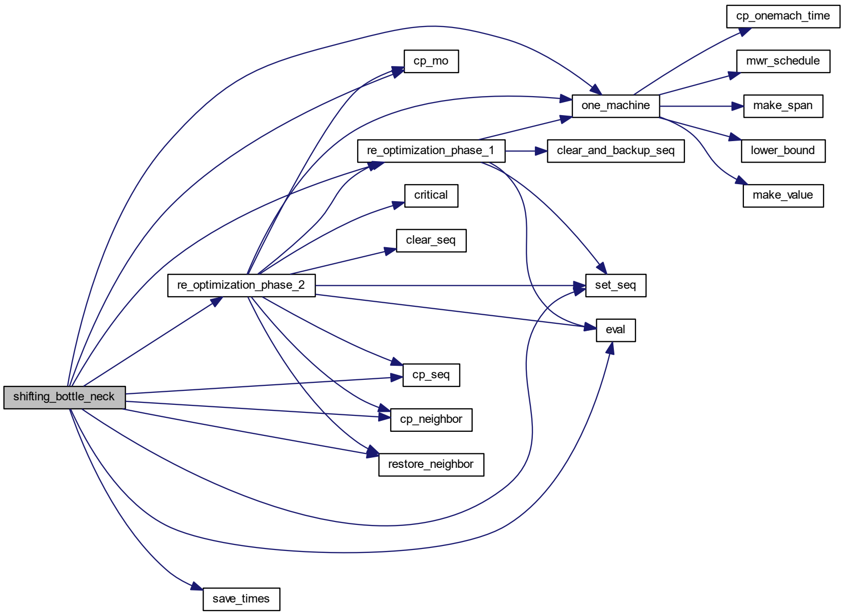
seq	The given sequence list. Will be updated when find a better makespan.
machine_order	Machine order.
try_time_set	Backtracing depth set.

Returns

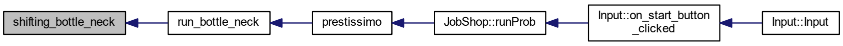
When the procedure is done. You should find the start time of the solution at the "start" field of the struct array job.

Definition at line 119 of file bottle.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.5 Variable Documentation

7.1.5.1 best_makespan

```
best_makespan = INFINITAS
```

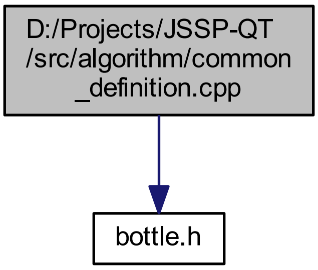
Store the best makespan value.

Definition at line 47 of file bottle.cpp.

7.2 D:/Projects/JSSP-QT/src/algorithm/common_definition.cpp File Reference

```
#include <bottle.h>
```

Include dependency graph for common_definition.cpp:



Variables

- job_t job [MAXJOB]
- int job_size = 1
- int machine_size = 1
- int terminate_flag = 0

7.2.1 Detailed Description

Author

Name1e5s

7.2.2 Variable Documentation

7.2.2.1 job

```
job
```

Data representation of all the jobs. All operations runs on this varible.

Definition at line 11 of file common_definition.cpp.

7.2.2.2 job_size

```
job_size = 1
```

Job number in this instance.

Definition at line 16 of file common_definition.cpp.

7.2.2.3 machine_size

```
machine_size = 1
```

Machine number in this instance.

Definition at line 21 of file common_definition.cpp.

7.2.2.4 terminate_flag

```
terminate_flag = 0
```

Should we stop???

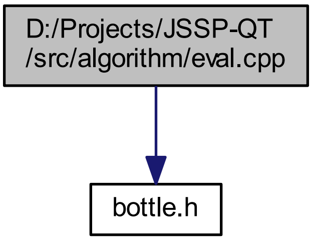
Definition at line 26 of file common_definition.cpp.

7.3 D:/Projects/JSSP-QT/src/algorithm/eval.cpp File Reference

makespan

```
#include <bottle.h>
```

Include dependency graph for eval.cpp:



Classes

- struct JOBMACHINEPAR

Typedefs

- typedef struct JOBMACHINEPAR job_machine_t

Functions

- int eval (sequence_t ∗seq)

Variables

- int magicnum = 0

7.3.1 Detailed Description

makespan

Function to calcuate makespan.

Author

Name1e5s

7.3.2 Typedef Documentation

7.3.2.1 job_machine_t

```
typedef struct JOBMACHINEPAR job_machine_t
```

Auxiliary struct to calcuate the makespan.

7.3.3 Function Documentation

7.3.3.1 eval()

```
int eval (
             sequence_t * seq )
```

Evaluate the makespan of the given sequence.

Parameters

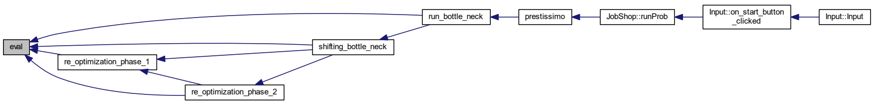
seq	The sequence of job.

Returns

The makespan of the sequence.

Definition at line 31 of file eval.cpp.

Here is the caller graph for this function:



7.3.4 Variable Documentation

7.3.4.1 magicnum

```
magicnum = 0
```

The number generated and managed by the God in the computer. Every one who changed the name of this feild will be seen as an evil and will be cursed by the God.

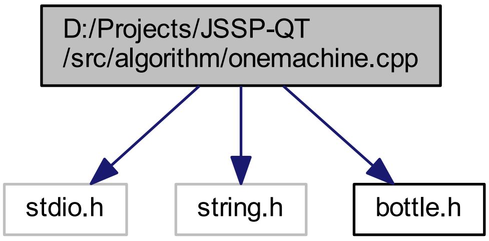
Definition at line 23 of file eval.cpp.

7.4 D:/Projects/JSSP-QT/src/algorithm/onemachine.cpp File Reference

One-machine sequencing.

```
#include <stdio.h>
#include <string.h>
#include <bottle.h>
```

Include dependency graph for onemachine.cpp:



Classes

- struct ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT

Macros

- #define ONEMACH_BBNODES 300

Typedefs

- typedef struct ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT onemach_bb_ass_t

Functions

- static void cp_onemach_time (onemach_times_t ∗mew, onemach_times_t ∗origin)
- static void mwr_schedule (onemach_times_t one, int ∗order)
- static int lower_bound (onemach_times_t one, int ∗jset, int jset_size)
- static int make_span (onemach_times_t one, int ∗order, int ∗jset, int ∗jset_size, int ∗cjob, int ∗pjob, int ∗make)
- static int make_value (onemach_times_t one, int ∗order)
- int one_machine (onemach_times_t one, int ∗bestorder)

7.4.1 Detailed Description

One-machine sequencing.

One-machine Sequencing Algorithm from Jacques Carlier

Author

TJenica

7.4.2 Macro Definition Documentation

7.4.2.1 ONEMACH_BBNODES

```
#define ONEMACH_BBNODES 300
```

Nodes number of the branch and bound tree to solve the one machine sequencing problem.

Definition at line 18 of file onemachine.cpp.

7.4.3 Typedef Documentation

7.4.3.1 onemach_bb_ass_t

```
typedef struct ONEMACHINE_BRANCH_AND_BOUND_ASSISTANT onemach_bb_ass_t
```

Info of node of a branch and bound tree.

7.4.4 Function Documentation

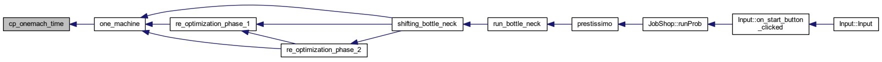
7.4.4.1 cp_onemach_time()

```
static void cp_onemach_time (
             onemach_times_t * mew,
             onemach_times_t * origin )  [inline], [static]
```

Copy origin onemach_times struct to mew By practice, change from memcpy to just write what we want to do is very important...

Definition at line 160 of file onemachine.cpp.

Here is the caller graph for this function:



7.4.4.2 lower_bound()

```
static int lower_bound (
             onemach_times_t one,
             int * job_set,
             int job_set_size )  [inline], [static]
```

Find the lower bound of the given machine on the given job order.

Parameters

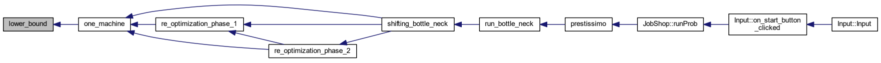
one	The representation of the given machine.
job_set	The set of job.
job_set_size	The size of job_set

Returns

Lowerbound of the machine. Which is just the sum of minimum estime and minimum mhtime and the sum of all the process time.

Definition at line 246 of file onemachine.cpp.

Here is the caller graph for this function:



7.4.4.3 make_span()

```
static int make_span (
             onemach_times_t one,
             int * order,
             int * job_set,
             int * job_set_size,
             int * critical_job_order,
             int * terminate_job_order,
             int * make )  [inline], [static]
```

Test if the job order is feasible and compute the make_span.

Parameters

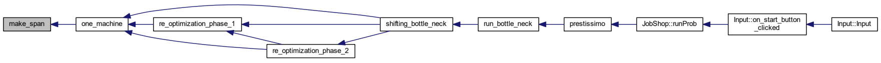
one	The representation of the given machine.
order	The given job order.
job_set	The set of job on the machine.
job_set_size	The size of job_set.
critical_job_order	
terminate_job_order	
make	The make_span.

Returns

If the order is OK return 1,else return 0.

Definition at line 278 of file onemachine.cpp.

Here is the caller graph for this function:



7.4.4.4 make_value()

```
static int make_value (
            onemach_times_t one,
            int * order )  [inline], [static]
```

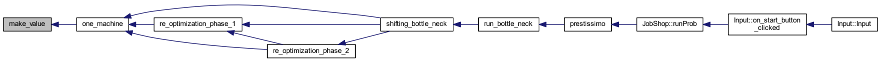
Compute the makespan of the given job order.

Parameters

one	Representation of the machine. order The given job order.

Definition at line 223 of file onemachine.cpp.

Here is the caller graph for this function:



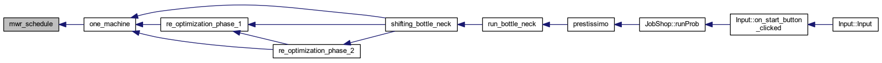
7.4.4.5 mwr_schedule()

```
static void mwr_schedule (
            onemach_times_t one,
            int * order )  [inline], [static]
```

Algorithm to find the most work remaining schedule by Schrage

Definition at line 171 of file onemachine.cpp.

Here is the caller graph for this function:



7.4.4.6 one_machine()

```
int one_machine (
            onemach_times_t one,
            int * bestorder )
```

The one-machine sequencing algorithm from "The one-machine sequencing problem" by Jacques Carlier.

Parameters

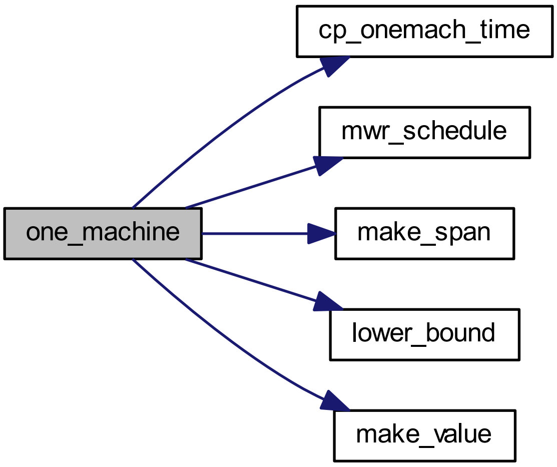
one	Representation of the machine.
bestorder	Best job order

Returns

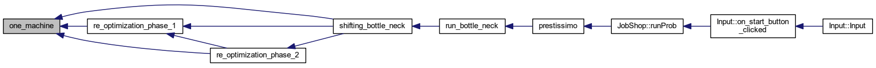
makespan

Definition at line 42 of file onemachine.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



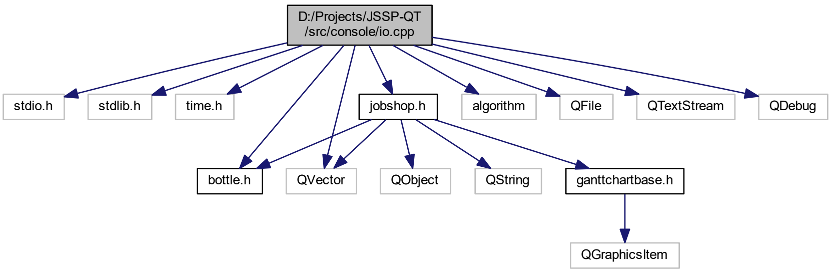
7.5 D:/Projects/JSSP-QT/src/console/io.cpp File Reference

IO.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <bottle.h>
#include <QVector>
```

```
#include <algorithm>
#include <jobshop.h>
#include <QFile>
#include <QTextStream>
#include <QDebug>
```

Include dependency graph for io.cpp:



Classes

- struct PAIR_ASSISTANT_TYPE

Typedefs

- typedef struct PAIR_ASSISTANT_TYPE pair_ass_t

Functions

- static void write_file (char ∗file_name, int ans, pair_ass_t ∗pairs, float times)
- int machine_sort_cmp (const void ∗a, const void ∗b)
- int starttime_sort_cmp (const void ∗a, const void ∗b)
- void prestissimo (void)

Variables

- float used_time
- int best_makespan

  Store the best makespan value.

7.5.1 Detailed Description

IO.

Functions to handle input and output.

Author

TJenica

7.5.2 Typedef Documentation

7.5.2.1 pair_ass_t

```
typedef struct PAIR_ASSISTANT_TYPE pair_ass_t
```

A temporary struct for converting the internal representation of the solution to the format required by those soreheads.

7.5.3 Function Documentation

7.5.3.1 machine_sort_cmp()

```
int machine_sort_cmp (
            const void * a,
            const void * b )
```

Function to compare machine number of two pairs for qsort.

Parameters

a	The first pair.
b	The second pair.

Returns

If machine number of a is lesser than b, then return a positive value, else return a non-positive value.

Definition at line 48 of file io.cpp.

7.5.3.2 prestissimo()

```
void prestissimo (
            void )
```

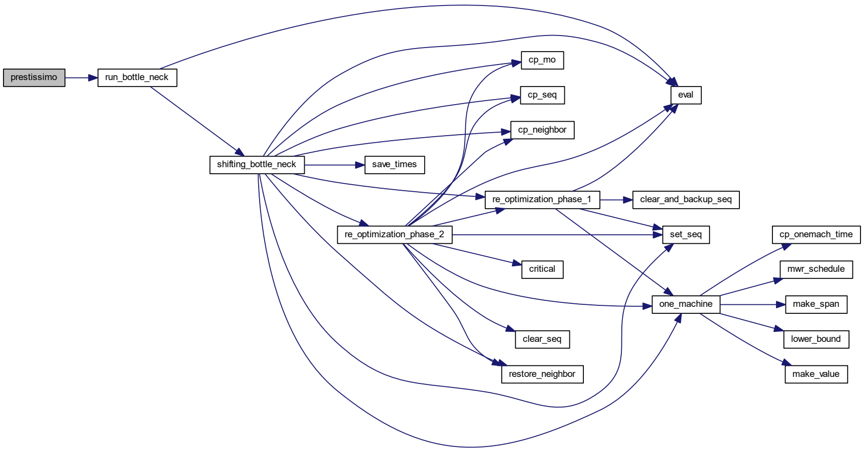
Convert internal solution representation structure to the format required by those nitpickers and print it.

Parameters

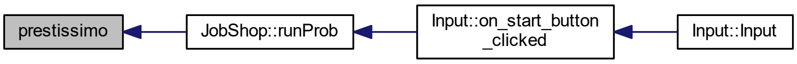
filename	Instance file path

Definition at line 70 of file io.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.3 starttime_sort_cmp()

```
int starttime_sort_cmp (
          const void * a,
          const void * b )
```

Function to compare start time of two pairs for qsort.

Parameters

a	The first pair.
b	The second pair.

Returns

If start time of a is lesser than b, then return a positive value, else return a non-positive value.

Definition at line 60 of file io.cpp.

7.5.3.4 write_file()

```
static void write_file (
             char * file_name,
             int ans,
             pair_ass_t * pairs,
             float times )  [inline], [static]
```

7.5.4 Variable Documentation

7.5.4.1 best_makespan

```
int best_makespan
```

Store the best makespan value.

Definition at line 47 of file bottle.cpp.

7.5.4.2 used_time

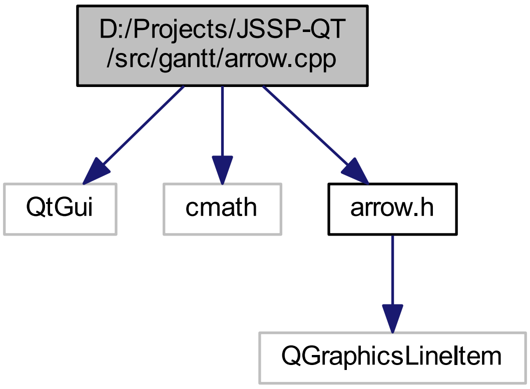
```
float used_time
```

Definition at line 32 of file io.cpp.

7.6 D:/Projects/JSSP-QT/src/gantt/arrow.cpp File Reference

```
#include <QtGui>
#include <cmath>
#include <arrow.h>
```

Include dependency graph for arrow.cpp:



Macros

- #define USE_MATH_DEFINES

Variables

- const qreal Pi = M_PI
- const qreal arrowSize = 10

7.6.1 Detailed Description

Draw arrow.

Author

Name1e5s

7.6.2 Macro Definition Documentation

7.6.2.1 USE_MATH_DEFINES

```
#define USE_MATH_DEFINES
```

Definition at line 10 of file arrow.cpp.

7.6.3 Variable Documentation

7.6.3.1 arrowSize

```
const qreal arrowSize = 10
```

Definition at line 16 of file arrow.cpp.

7.6.3.2 Pi

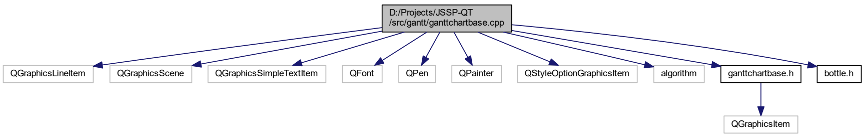
```
const qreal Pi = M_PI
```

Definition at line 15 of file arrow.cpp.

7.7 D:/Projects/JSSP-QT/src/gantt/ganttchartbase.cpp File Reference

```
#include <QGraphicsLineItem>
#include <QGraphicsScene>
#include <QGraphicsSimpleTextItem>
#include <QFont>
#include <QPen>
#include <QPainter>
#include <QStyleOptionGraphicsItem>
#include <algorithm>
#include <ganttchartbase.h>
#include <bottle.h>
```

Include dependency graph for ganttchartbase.cpp:



7.7.1 Detailed Description

Basic definition of our gantt chart.

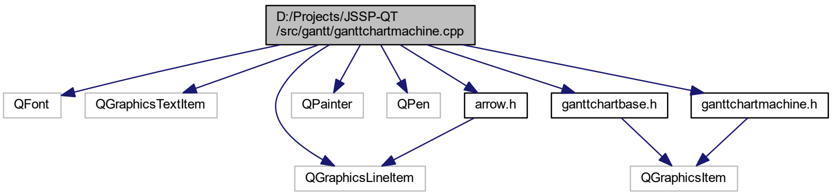
Author

Name1e5s

7.8 D:/Projects/JSSP-QT/src/gantt/ganttchartmachine.cpp File Reference

```
#include <QFont>
#include <QGraphicsTextItem>
#include <QGraphicsLineItem>
#include <QPainter>
#include <QPen>
#include <arrow.h>
#include <ganttchartbase.h>
#include <ganttchartmachine.h>
```

Include dependency graph for ganttchartmachine.cpp:



7.8.1 Detailed Description

Draw machine for our gantt chart.

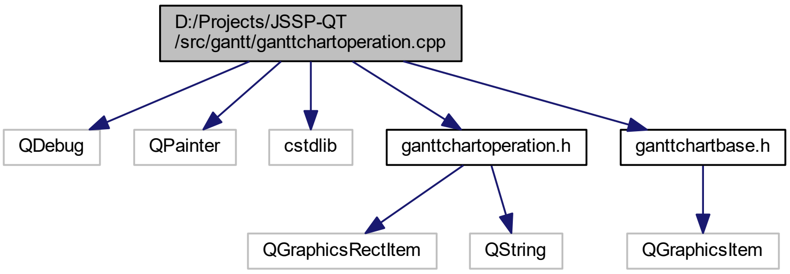
Author

Name1e5s

7.9 D:/Projects/JSSP-QT/src/gantt/ganttchartoperation.cpp File Reference

```
#include <QDebug>
#include <QPainter>
#include <cstdlib>
#include <ganttchartoperation.h>
#include <ganttchartbase.h>
```

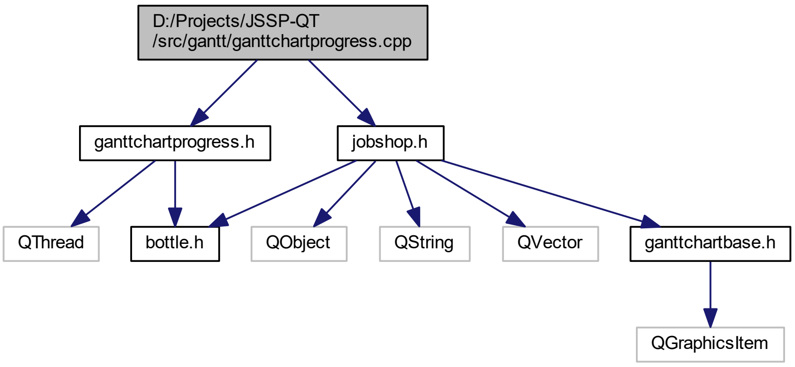
Include dependency graph for ganttchartoperation.cpp:



7.10 D:/Projects/JSSP-QT/src/gantt/ganttchartprogress.cpp File Reference

```
#include <ganttchartprogress.h>
#include <jobshop.h>
```

Include dependency graph for ganttchartprogress.cpp:



Variables

- int best_makespan

  Store the best makespan value.

7.10.1 Detailed Description

Author

Name1e5s

7.10.2 Variable Documentation

7.10.2.1 best_makespan

```
int best_makespan
```

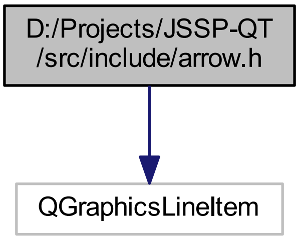
Store the best makespan value.

Definition at line 47 of file bottle.cpp.

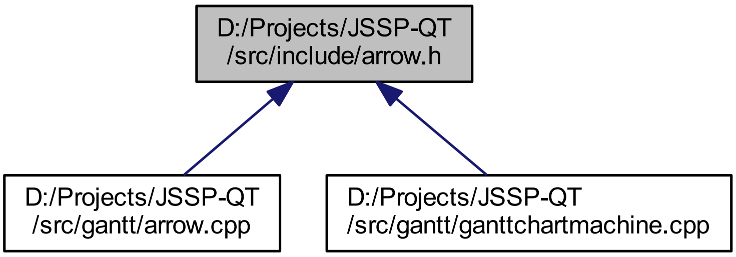
7.11 D:/Projects/JSSP-QT/src/include/arrow.h File Reference

```
#include <QGraphicsLineItem>
```

Include dependency graph for arrow.h:



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



Classes

- class Arrow

7.11.1 Detailed Description

This file defines how to draw an arrow on thegantt chart.

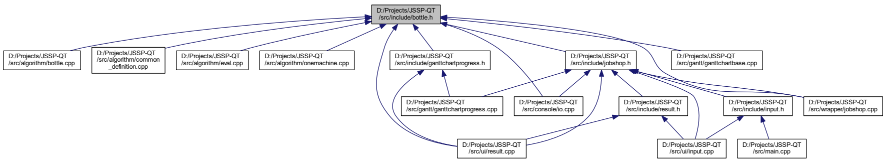
Author

Name1e5s

7.12 D:/Projects/JSSP-QT/src/include/bottle.h File Reference

Header file for the whole project.

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



Classes

- struct JOB
- struct SEQUENCE
- struct ONEMACHINEstime

Macros

- #define MAXJOB 30
- #define MAXMACHINE 30
- #define INFINITAS 0x7fffffff
- #define MAX(a, b) ((a) > (b) ? (a) : (b))

Typedefs

- typedef struct JOB job_t
- typedef struct SEQUENCE sequence_t
- typedef struct ONEMACHINEstime onemach_times_t

Functions

- void prestissimo (void)
- void run_bottle_neck (void)
- int one_machine (onemach_times_t one, int ∗bestorder)

Variables

- job_t job [MAXJOB]
- int job_size
- int machine_size
- int terminate_flag

7.12.1 Detailed Description

Header file for the whole project.

A Simple Old-fashion Implementation Of The Well-known Shifting Bottleneck Procedure For Job Shop Scheduling Problem(JSSP).The codes are based on "The Shifting Bottleneck Procedure for Job Shop Scheduling" by J. Adams et al.

Author

Name1e5s

7.12.2 Macro Definition Documentation

7.12.2.1 INFINITAS

```
#define INFINITAS 0x7fffffff
```

A integer that can be seen as infinity – should be bigger than the biggest makespan of all the instances. Hence, 0x7fffffff (a.k.a INT_MAX) is a good choice

Definition at line 31 of file bottle.h.

7.12.2.2 MAX

```
#define MAX(
              a,
              b ) ((a) > (b) ? (a) : (b))
```

A regular macro that returns the bigger value bewteen a and b.

Definition at line 36 of file bottle.h.

7.12.2.3 MAXJOB

```
#define MAXJOB 30
```

The most jobs this program can handle.

Definition at line 19 of file bottle.h.

7.12.2.4 MAXMACHINE

```
#define MAXMACHINE 30
```

The most machines this program can handle.

Definition at line 24 of file bottle.h.

7.12.3 Typedef Documentation

7.12.3.1 job_t

```
typedef struct JOB job_t
```

Data representation for a job.

7.12.3.2 onemach_times_t

```
typedef struct ONEMACHINEstime onemach_times_t
```

Store the time info for every job runs on the same machine.

7.12.3.3 sequence_t

```
typedef struct SEQUENCE sequence_t
```

Job sequences on a machine.

7.12.4 Function Documentation

7.12.4.1 one_machine()

```
int one_machine (
            onemach_times_t one,
            int * bestorder )
```

The one-machine sequencing algorithm from "The one-machine sequencing problem" by Jacques Carlier.

Parameters

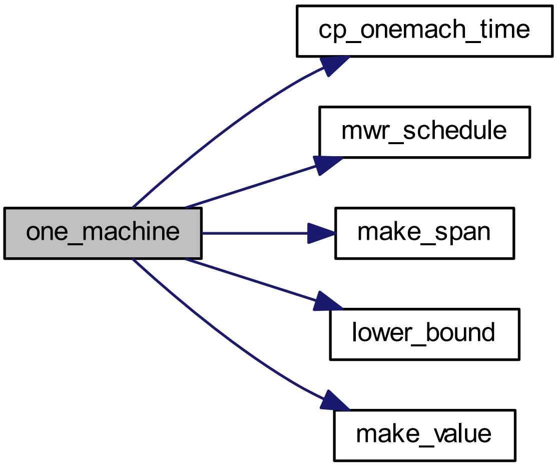
one	Representation of the machine.
bestorder	Best job order

Returns

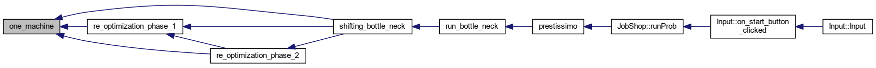
makespan

Definition at line 42 of file onemachine.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



7.12.4.2 prestissimo()

```
void prestissimo (
             void  )
```

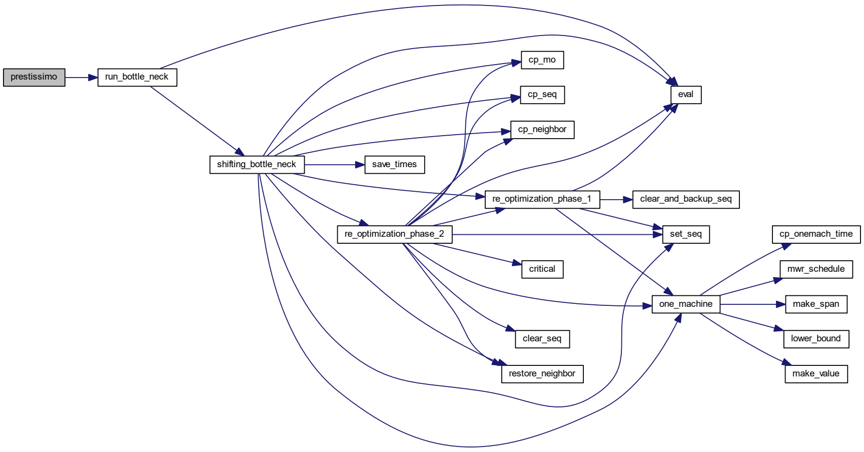
Convert internal solution representation structure to the format required by those nitpickers and print it.

Parameters

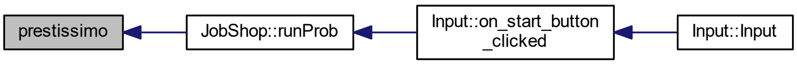
filename	Instance file path

Definition at line 70 of file io.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



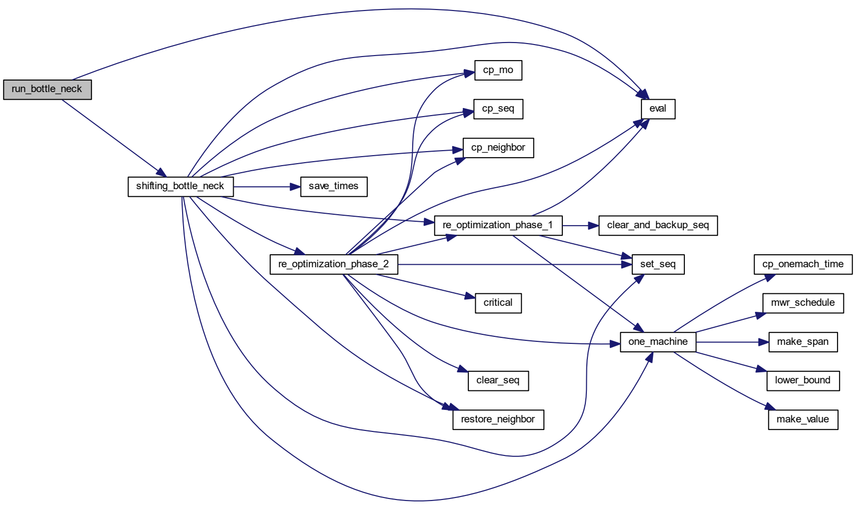
7.12.4.3 run_bottle_neck()

```
void run_bottle_neck (
            void  )
```

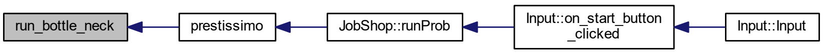
Driver of the Shifting Bottleneck Procedure We can change here to have a balance bewteen run time and makespan...

Definition at line 67 of file bottle.cpp.

Here is the call graph for this function:



Here is the caller graph for this function:



7.12.5 Variable Documentation

7.12.5.1 job

```
job_t job[MAXJOB]
```

Data representation of all the jobs. All operations runs on this varible.

Definition at line 11 of file common_definition.cpp.

7.12.5.2 job_size

```
int job_size
```

Job number in this instance.

Definition at line 16 of file common_definition.cpp.

7.12.5.3 machine_size

```
int machine_size
```

Machine number in this instance.

Definition at line 21 of file common_definition.cpp.

7.12.5.4 terminate_flag

```
int terminate_flag
```

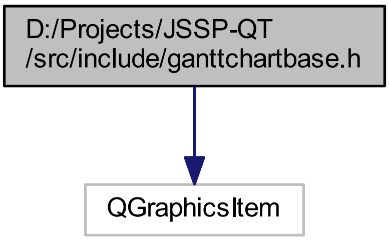
Should we stop???

Definition at line 26 of file common_definition.cpp.

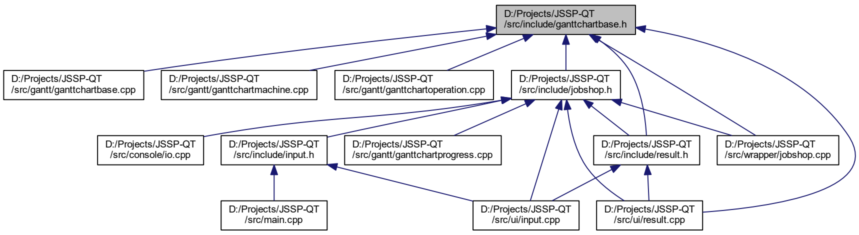
7.13 D:/Projects/JSSP-QT/src/include/ganttchartbase.h File Reference

```
#include <QGraphicsItem>
```

Include dependency graph for ganttchartbase.h:



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



Classes

- class GanttChartBase

7.13.1 Detailed Description

This file contains the base class of the gantt chart.

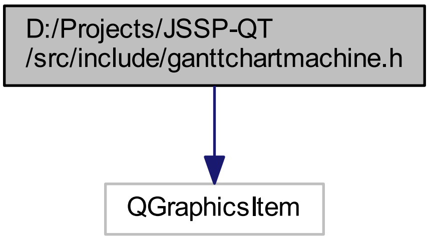
Author

Name1e5s

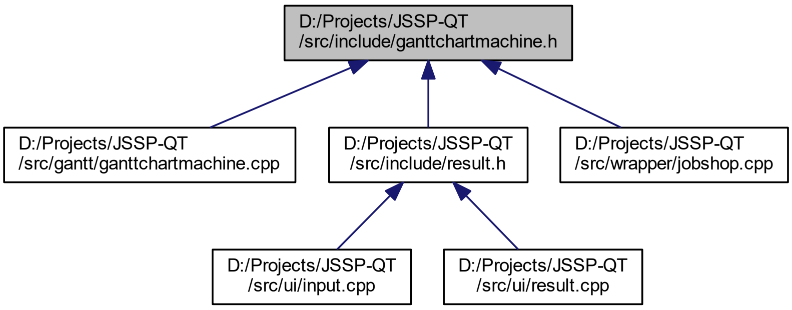
7.14 D:/Projects/JSSP-QT/src/include/ganttchartmachine.h File Reference

```
#include <QGraphicsItem>
```

Include dependency graph for ganttchartmachine.h:



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


D:/Projects/JSSP-QT /src/include/ganttchartmachine.h
D:/Projects/JSSP-QT /src/gantt/ganttchartmachine.cpp
D:/Projects/JSSP-QT /src/include/result.h
D:/Projects/JSSP-QT /src/wrapper/jobshop.cpp
D:/Projects/JSSP-QT /src/ui/input.cpp
D:/Projects/JSSP-QT /src/ui/result.cpp

Classes

- class GanttChartMachine

7.14.1 Detailed Description

This file defines how to present the gantt chart per machine.

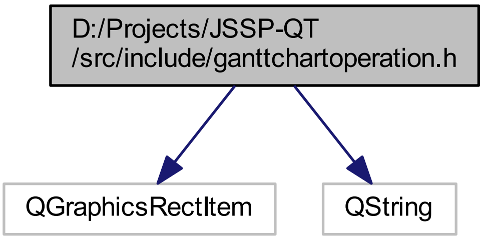
Author

Name1e5s

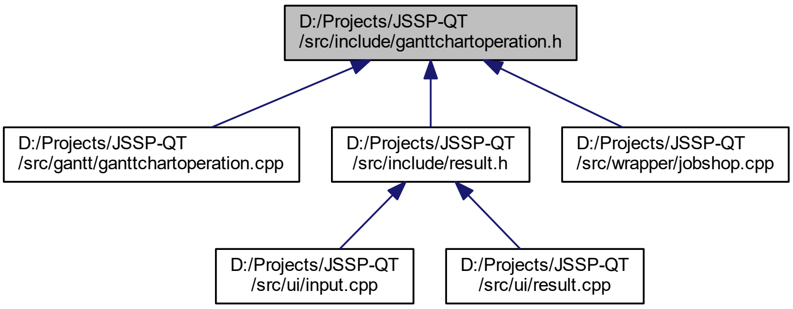
7.15 D:/Projects/JSSP-QT/src/include/ganttchartoperation.h File Reference

```
#include <QGraphicsRectItem>
#include <QString>
```

Include dependency graph for ganttchartoperation.h:



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



Classes

- class GanttChartOperation

7.15.1 Detailed Description

This file defines how to present a operation on the m=gantt chart.

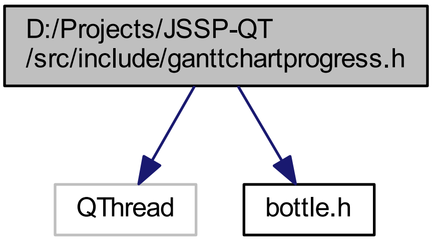
Author

Name1e5s

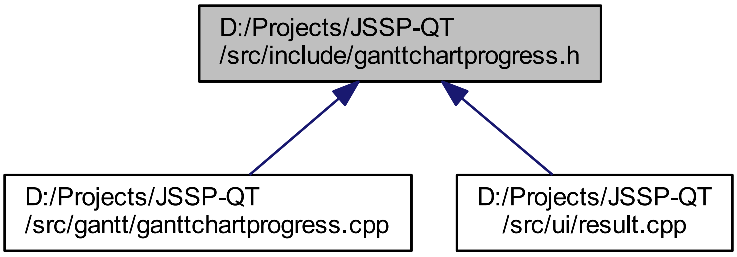
7.16 D:/Projects/JSSP-QT/src/include/ganttchartprogress.h File Reference

```
#include <QThread>
#include <bottle.h>
```

Include dependency graph for ganttchartprogress.h:



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



Classes

- class GanttChartProgress

Variables

- int best_makespan

  Store the best makespan value.

7.16.1 Detailed Description

Author

Name1e5s

7.16.2 Variable Documentation

7.16.2.1 best_makespan

```
int best_makespan
```

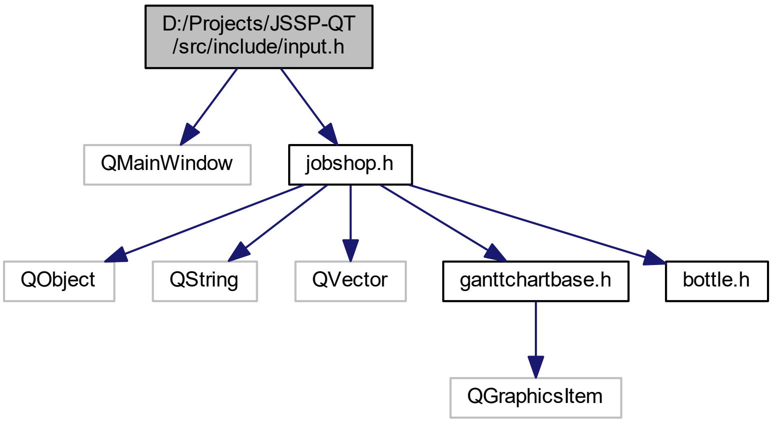
Store the best makespan value.

Definition at line 47 of file bottle.cpp.

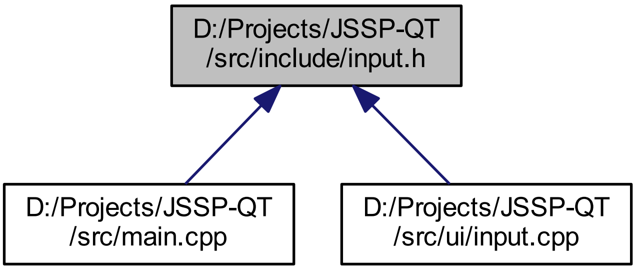
7.17 D:/Projects/JSSP-QT/src/include/input.h File Reference

```
#include <QMainWindow>
#include <jobshop.h>
```

Include dependency graph for input.h:



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


D:/Projects/JSSP-QT /src/include/input.h
D:/Projects/JSSP-QT /src/main.cpp
D:/Projects/JSSP-QT /src/ui/input.cpp

Classes

- class Input

Namespaces

- Ui

7.17.1 Detailed Description

This file defines the input window.

Author

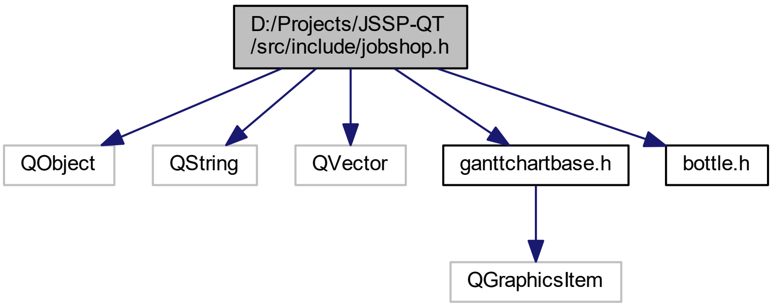
Name1e5s

7.18 D:/Projects/JSSP-QT/src/include/jobshop.h File Reference

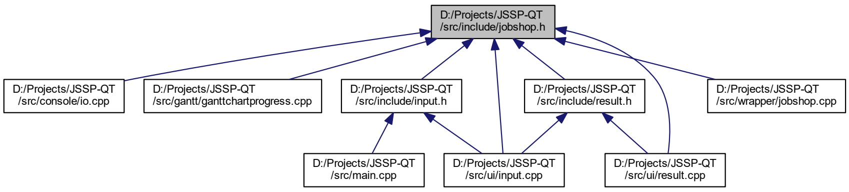
```
#include <QObject>
#include <QString>
#include <QVector>
#include <ganttchartbase.h>
```

`#include <bottle.h>`

Include dependency graph for jobshop.h:



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



Classes

- struct Fixer
- class JobShop

7.18.1 Detailed Description

This file is a simple wrapper for the previous CLI version of this project to make it compatible with QT.

!!!Note: This implementation is not a good pratice.

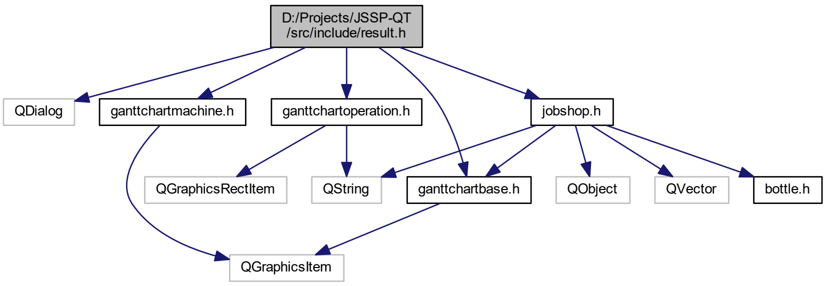
Author

Name1e5s

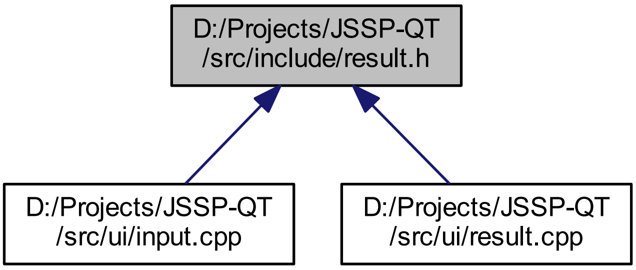
7.19 D:/Projects/JSSP-QT/src/include/result.h File Reference

```
#include <QDialog>
#include <ganttchartbase.h>
#include <ganttchartmachine.h>
#include <ganttchartoperation.h>
#include <jobshop.h>
```

Include dependency graph for result.h:



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


D:/Projects/JSSP-QT /src/include/result.h
D:/Projects/JSSP-QT /src/ui/input.cpp
D:/Projects/JSSP-QT /src/ui/result.cpp

Classes

- class Result

Namespaces

- Ui

7.19.1 Detailed Description

This file defines a simple dialog to show our Gantt chart and it allows user to fix a machins at a given time. The format of the fix command is: $[time] [machine to be fixed] [duration]

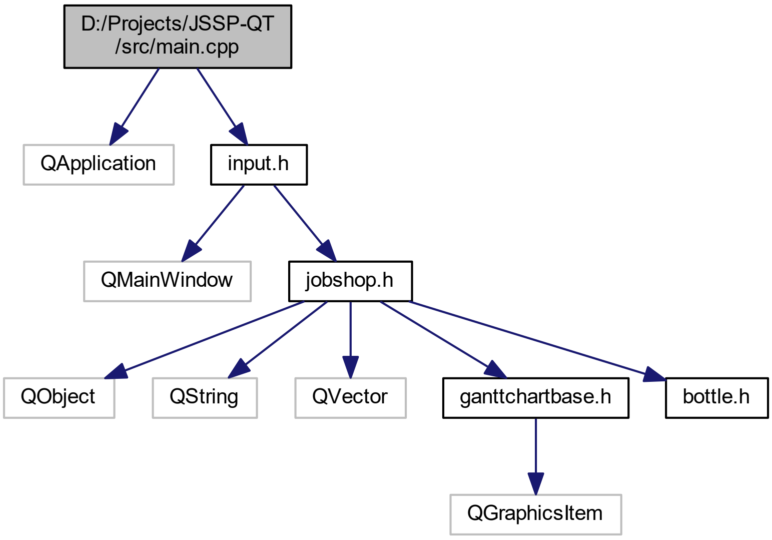
Author

Name1e5s

7.20 D:/Projects/JSSP-QT/src/main.cpp File Reference

```
#include <QApplication>
#include <input.h>
```

Include dependency graph for main.cpp:



Functions

- int main (int ac, char ∗av[ ])

7.20.1 Detailed Description

Enterpoint of the program.

Author

Name1e5s

7.20.2 Function Documentation

7.20.2.1 main()

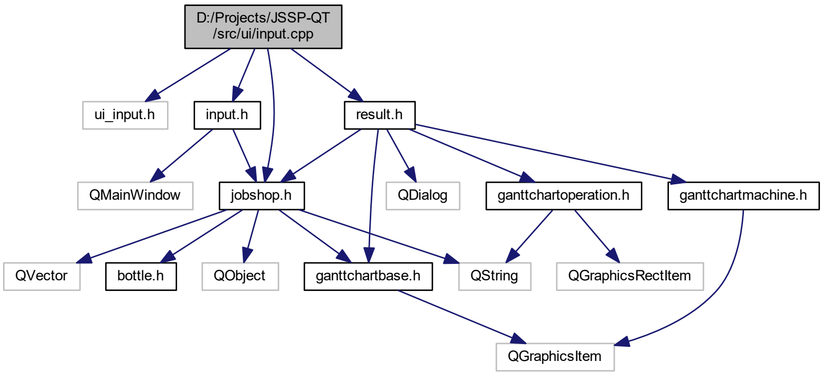
```
int main (
            int ac,
            char * av[] )
```

Definition at line 11 of file main.cpp.

7.21 D:/Projects/JSSP-QT/src/ui/input.cpp File Reference

```
#include "ui_input.h"
#include <input.h>
#include <result.h>
#include <jobshop.h>
```

Include dependency graph for input.cpp:



7.21.1 Detailed Description

Draw input window.

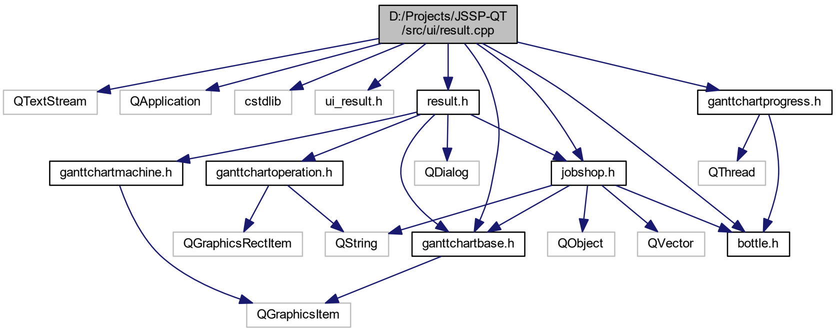
Author

Name1e5s

7.22 D:/Projects/JSSP-QT/src/ui/result.cpp File Reference

```
#include <QTextStream>
#include <QApplication>
#include <cstdlib>
#include "ui_result.h"
#include <result.h>
#include <ganttchartprogress.h>
#include <ganttchartbase.h>
#include <bottle.h>
#include <jobshop.h>
```

Include dependency graph for result.cpp:



Classes

- struct pair

  The pair struct for sorting machines.

Functions

- int starttime_cmp (const void ∗a, const void ∗b)

  Function to compare starttime of two pairs for qsort.

Variables

- int best_makespan

  Store the best makespan value.

7.22.1 Detailed Description

Draw result dialog.

Author

Name1e5s

7.22.2 Function Documentation

7.22.2.1 starttime_cmp()

```
int starttime_cmp (
             const void * a,
             const void * b )
```

Function to compare starttime of two pairs for qsort.

Parameters

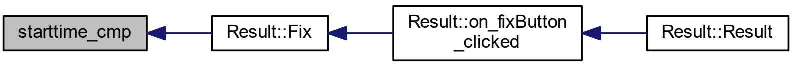
a	The first pair.
b	The second pair.

Returns

If start time of a is lesser than b, then return a positive value, else return a non-positive value.

Definition at line 39 of file result.cpp.

Here is the caller graph for this function:



7.22.3 Variable Documentation

7.22.3.1 best_makespan

```
int best_makespan
```

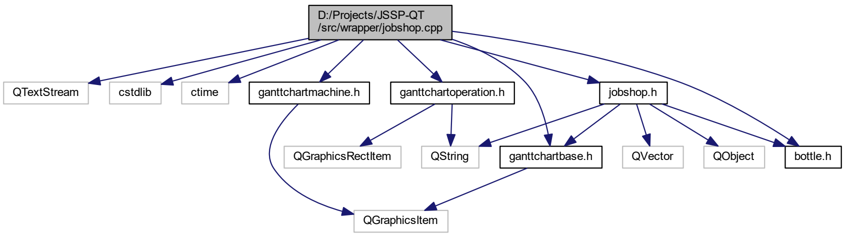
Store the best makespan value.

Definition at line 47 of file bottle.cpp.

7.23 D:/Projects/JSSP-QT/src/wrapper/jobshop.cpp File Reference

```
#include <QTextStream>
#include <cstdlib>
#include <ctime>
#include <ganttchartbase.h>
#include <ganttchartmachine.h>
#include <ganttchartoperation.h>
#include <bottle.h>
#include <jobshop.h>
```

Include dependency graph for jobshop.cpp:



Variables

- int best_makespan

  Store the best makespan value.

7.23.1 Detailed Description

Simple wrapper of previous version of JSSP solver.

Author

Name1e5s

7.23.2 Variable Documentation

7.23.2.1 best_makespan

```
int best_makespan
```

Store the best makespan value.

Definition at line 47 of file bottle.cpp.

Index