

Greedy

Generated by Doxygen 1.9.6



<b>1 STVD-PMS</b>	<b>1</b>
1.1 Data loader	1
1.2 Natural Language Text Processing with XML	1
<b>2 Namespace Index</b>	<b>3</b>
2.1 Namespace List	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Namespace Documentation</b>	<b>9</b>
5.1 1_generate_latency Namespace Reference	9
5.1.1 Function Documentation	9
5.1.1.1 check_timestamp_valid()	9
5.1.1.2 checking()	9
5.1.1.3 convert_xlm_file_to_date()	10
5.1.1.4 main()	10
5.1.1.5 nomarlizeText()	10
5.1.1.6 read_xml()	10
5.1.2 Variable Documentation	10
5.1.2.1 data_list	10
5.2 common Namespace Reference	10
5.2.1 Function Documentation	11
5.2.1.1 check_file()	11
5.2.1.2 get_channel_name_by_code()	11
5.2.1.3 map_channel()	11
<b>6 Class Documentation</b>	<b>13</b>
6.1 Greedy Class Reference	13
6.1.1 Constructor & Destructor Documentation	13
6.1.1.1 Greedy() [1/2]	14
6.1.1.2 Greedy() [2/2]	14
6.1.2 Member Function Documentation	14
6.1.2.1 exchangeCheck()	14
6.1.2.2 findCi()	14
6.1.2.3 findPoints()	15
6.1.2.4 findQi()	15
6.1.2.5 setAlpha()	15
6.1.2.6 setMaxChannel()	15
6.1.2.7 setOmega()	16
6.1.2.8 solve()	16

6.2 Interval Class Reference	16
6.2.1 Constructor & Destructor Documentation	17
6.2.1.1 Interval() [1/2]	17
6.2.1.2 Interval() [2/2]	17
6.2.2 Member Function Documentation	17
6.2.2.1 generateRandomWeight()	18
6.2.2.2 getChannel()	18
6.2.2.3 getHashCode()	18
6.2.2.4 getId()	19
6.2.2.5 getStartDate()	19
6.2.2.6 getStopDate()	19
6.2.2.7 getWeight()	19
6.2.2.8 printInterval()	20
6.2.2.9 setStartDate()	20
6.2.2.10 setStopDate()	20
6.3 IntervalParser Class Reference	20
6.3.1 Constructor & Destructor Documentation	21
6.3.1.1 IntervalParser()	21
6.3.1.2 ~IntervalParser()	21
6.3.2 Member Function Documentation	21
6.3.2.1 originToZero()	21
6.3.2.2 parse()	22
6.3.2.3 sortIntervals()	22
6.4 Point Class Reference	22
6.4.1 Constructor & Destructor Documentation	23
6.4.1.1 Point() [1/2]	23
6.4.1.2 Point() [2/2]	23
6.4.2 Member Function Documentation	23
6.4.2.1 getCost()	23
6.4.2.2 getId()	24
6.4.2.3 getMinInterval()	24
6.4.2.4 setCost()	24
6.4.2.5 setMinInterval()	24
6.5 SolutionParser Class Reference	25
6.5.1 Constructor & Destructor Documentation	25
6.5.1.1 SolutionParser()	25
6.5.1.2 ~SolutionParser()	25
6.5.2 Member Function Documentation	25
6.5.2.1 writeCSV()	25
<b>7 File Documentation</b>	<b>27</b>
7.1 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Greedy.cpp File Reference	27

7.2 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Greedy.h File Reference . . . . .	27
7.3 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Greedy.h . . . . .	27
7.4 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Interval.cpp File Reference . . . . .	28
7.5 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Interval.h File Reference . . . . .	28
7.6 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Interval.h . . . . .	28
7.7 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/IntervalParser.cpp File Reference . .	29
7.8 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/IntervalParser.h File Reference . . .	29
7.8.1 Macro Definition Documentation . . . . .	29
7.8.1.1 IntervalParser8H . . . . .	30
7.9 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/IntervalParser.h . . . . .	30
7.10 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/main.cpp File Reference . . . . .	30
7.10.1 Function Documentation . . . . .	30
7.10.1.1 main() . . . . .	31
7.11 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Point.cpp File Reference . . . . .	31
7.12 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Point.h File Reference . . . . .	31
7.13 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Point.h . . . . .	31
7.14 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Python/1_generate_latency.py File Reference . . . . .	31
7.15 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Python/common.py File Reference	32
7.16 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Python/README.md File Reference	32
7.17 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/SolutionParser.cpp File Reference .	32
7.18 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/SolutionParser.h File Reference . .	32
7.19 E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/SolutionParser.h . . . . .	33
<b>Index</b>	<b>35</b>



# Chapter 1

## STVD-PMS

This repo is used to extract the metadata from TV channels

### 1.1 Data loader

Download data from the link: <https://drive.google.com/drive/folders/1wvL6HJwhUpC14QFqRUHy1WuobupTNRcR1?usp=sharing>

Move the folder 'data' into the project directory.

### 1.2 Natural Language Text Processing with XML

The first step is to extract all of the TV programs during from at 06:00 AM to 02:00 AM of a next day. However, the TV program could be error-prone due to the nameing convention and it requires some filter rules as follows:

- Ignore the program having duration is less then five minutes (<5 mins>);
- Filter out any two programs are overlapped.

Sepecifically, given a list of the xml files, the collection.csv file will be generated as following.

- Input: the list of XML file with filenames formatted yyyy-mm-dd\_HHMMSS.xml
- Ouput: The list of TV programs that provided for each day based on the name of the XML file.

For example:

Input: The file 2022-01-01\_100000.xml

Ouput: List of TV programs which its starts FROM at 06:00:00 01-01-2022 TO at 02:00:00 02-01-2022

Notes: Using SHA-1 hash function for robustness of TV titles.





## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">1_generate_latency</a> . . . . .	9
<a href="#">common</a> . . . . .	10



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Greedy</a>	13
<a href="#">Interval</a>	16
<a href="#">IntervalParser</a>	20
<a href="#">Point</a>	22
<a href="#">SolutionParser</a>	25



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Greedy.cpp . . . . .	27
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Greedy.h . . . . .	27
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Interval.cpp . . . . .	28
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Interval.h . . . . .	28
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/IntervalParser.cpp . . . . .	29
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/IntervalParser.h . . . . .	29
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/main.cpp . . . . .	30
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Point.cpp . . . . .	31
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Point.h . . . . .	31
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/SolutionParser.cpp . . . . .	32
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/SolutionParser.h . . . . .	32
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Python/1_generate_latency.py . . . . .	31
E:/Poly/DI5/PRD/Algo_workspace/GreedyAlpha/Greedy/Greedy/Python/common.py . . . . .	32



## Chapter 5

# Namespace Documentation

### 5.1 1\_generate\_latency Namespace Reference

#### Functions

- def [nomarlizeText](#) (element)
- def [check\\_timestamp\\_valid](#) (filename\_str, date\_valid)
- def [convert\\_xlm\\_file\\_to\\_date](#) (filename\_str)
- def [checking](#) ()
- def [read\\_xml](#) (base\_path, input\_file, channel\_c)
- def [main](#) ()

#### Variables

- list [data\\_list](#) = []

#### 5.1.1 Function Documentation

##### 5.1.1.1 [check\\_timestamp\\_valid\(\)](#)

```
def 1_generate_latency.check_timestamp_valid (
    filename_str,
    date_valid )
```

##### 5.1.1.2 [checking\(\)](#)

```
def 1_generate_latency.checking ( )
```

#### 5.1.1.3 `convert_xlm_file_to_date()`

```
def l_generate_latency.convert_xlm_file_to_date (
    filename_str )
```

#### 5.1.1.4 `main()`

```
def l_generate_latency.main ( )
```

#### 5.1.1.5 `nomarlizeText()`

```
def l_generate_latency.nomarlizeText (
    element )
```

#### 5.1.1.6 `read_xml()`

```
def l_generate_latency.read_xml (
    base_path,
    input_file,
    channel_c )
```

### 5.1.2 Variable Documentation

#### 5.1.2.1 `data_list`

```
list l_generate_latency.data_list = [ ]
```

## 5.2 common Namespace Reference

### Functions

- def [get\\_channel\\_name\\_by\\_code](#) (channel\_code)
- def [check\\_file](#) (file\_name, path\_name)
- def [map\\_channel](#) (channel\_code)



## 5.2.1 Function Documentation

### 5.2.1.1 `check_file()`

```
def common.check_file (
    file_name,
    path_name )
```

### 5.2.1.2 `get_channel_name_by_code()`

```
def common.get_channel_name_by_code (
    channel_code )
```

### 5.2.1.3 `map_channel()`

```
def common.map_channel (
    channel_code )
```



## Chapter 6

# Class Documentation

### 6.1 Greedy Class Reference

```
#include <Greedy.h>
```

#### Public Member Functions

- [Greedy](#) ()  
*Construct a new [Greedy](#):: [Greedy](#) object.*
- [Greedy](#) (vector< [Interval](#) > omega, short maxChannel, float alpha)  
*Construct a new [Greedy](#):: [Greedy](#) object.*
- void [setOmega](#) (vector< [Interval](#) > omega)  
*Sets a new list of intervals to find a solution for.*
- void [setMaxChannel](#) (short maxChannel)  
*Sets a new max channel.*
- void [setAlpha](#) (float alpha)  
*Sets a new alpha value.*
- void [findQi](#) (int omega\_id)  
*Finds the Qi list of intervals (every interval that intersects with the current interval)*
- void [findPoints](#) ()  
*Finds and initiates the points of the Qi list.*
- void [findCi](#) ()  
*Finds the Ci list of intervals (minimal cost of the Qi list)*
- bool [exchangeCheck](#) ([Interval](#) \*i)  
*Checks if the exchange of Ci' (Ci + i') by i is worth it.*
- vector< [Interval](#) \* > [solve](#) ()

#### 6.1.1 Constructor & Destructor Documentation

**6.1.1.1 Greedy()** [1/2]

```
Greedy::Greedy ( )
```

Construct a new [Greedy::Greedy](#) object.

< Alpha value

**6.1.1.2 Greedy()** [2/2]

```
Greedy::Greedy (
    vector< Interval > omega,
    short maxChannel,
    float alpha )
```

Construct a new [Greedy::Greedy](#) object.

**Parameters**

<i>omega</i>	List of all intervals
--------------	-----------------------

**6.1.2 Member Function Documentation****6.1.2.1 exchangeCheck()**

```
bool Greedy::exchangeCheck (
    Interval * i )
```

Checks if the exchange of Ci' (Ci + i') by i is worth it.

**Returns**

true If the exchange is worth it  
false If the exchange is not worth it

**6.1.2.2 findCi()**

```
void Greedy::findCi ( )
```

Finds the Ci list of intervals (minimal cost of the Qi list)

## Parameters

<i>point</i> ↔ <i>Id</i>	Id of the point in the points list
-----------------------------	------------------------------------

**6.1.2.3 findPoints()**

```
void Greedy::findPoints ( )
```

Finds and initiates the points of the Qi list.

**6.1.2.4 findQi()**

```
void Greedy::findQi (
    int omega_id )
```

Finds the Qi list of intervals (every interval that intersects with the current interval)

## Parameters

<i>omega</i> ↔ <i>_id</i>	Id of the interval in the omega list
------------------------------	--------------------------------------

**6.1.2.5 setAlpha()**

```
void Greedy::setAlpha (
    float alpha )
```

Sets a new alpha value.

## Parameters

<i>alpha</i>	
--------------	--

**6.1.2.6 setMaxChannel()**

```
void Greedy::setMaxChannel (
    short maxChannel )
```

Sets a new max channel.

## Parameters

<i>maxChannel</i>	
-------------------	--

**6.1.2.7 setOmega()**

```
void Greedy::setOmega (
    vector< Interval > omega )
```

Sets a new list of intervals to find a solution for.

## Parameters

<i>omega</i>	
--------------	--

**6.1.2.8 solve()**

```
vector< Interval * > Greedy::solve ( )
```

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

- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[Greedy.h](#)
- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[Greedy.cpp](#)

**6.2 Interval Class Reference**

```
#include <Interval.h>
```

**Public Member Functions**

- [Interval](#) ()  
*Construct a new [Interval](#)::[Interval](#) object.*
- [Interval](#) (short id, long long startDate, long long stopDate, string channel, string hashCode)  
*Construct a new [Interval](#)::[Interval](#) object.*
- short [getId](#) ()  
*Gets the id of the interval.*
- long long [getStartDate](#) ()  
*Gets the start date of the interval.*
- long long [getStopDate](#) ()  
*Gets the stop date of the interval.*
- string [getChannel](#) ()  
*Gets the channel of the interval.*

- string [getHashCode](#) ()  
*Gets the hashcode of the interval.*
- short [getWeight](#) ()  
*Gets the weight of the interval.*
- void [setStartDate](#) (long long startDate)  
*Gets the point list of the interval.*
- void [setStopDate](#) (long long stopDate)  
*Sets the stop date of the interval.*
- void [generateRandomWeight](#) ()  
*Sets the point list of the interval.*
- void [printInterval](#) ()  
*Prints the interval.*

## 6.2.1 Constructor & Destructor Documentation

### 6.2.1.1 Interval() [1/2]

```
Interval::Interval ( )
```

Construct a new [Interval::Interval](#) object.

### 6.2.1.2 Interval() [2/2]

```
Interval::Interval (
    short id,
    long long startDate,
    long long stopDate,
    string channel,
    string hashcode )
```

Construct a new [Interval::Interval](#) object.

#### Parameters

<i>id</i>	<a href="#">Interval</a> id
<i>startDate</i>	<a href="#">Interval</a> start date
<i>stopDate</i>	<a href="#">Interval</a> stop date
<i>channel</i>	<a href="#">Interval</a> channel
<i>hashcode</i>	<a href="#">Interval</a> hashcode

## 6.2.2 Member Function Documentation

### 6.2.2.1 generateRandomWeight()

```
void Interval::generateRandomWeight ( )
```

Sets the point list of the interval.

#### Parameters

<i>pointList</i>	<a href="#">Point</a> list of the interval
------------------	--

Adds a point to the point list of the interval

#### Parameters

<i>point</i>	<a href="#">Point</a> to add
--------------	------------------------------

Gets the first point of the interval

#### Returns

short First point of the interval

Gets the last point of the interval

#### Returns

short Last point of the interval

Generates a random weight for the interval using its hashcode

### 6.2.2.2 getChannel()

```
string Interval::getChannel ( )
```

Gets the channel of the interval.

#### Returns

short Channel of the interval

### 6.2.2.3 getHashCode()

```
string Interval::getHashCode ( )
```

Gets the hashcode of the interval.

#### Returns

char\* Hashcode of the interval



#### 6.2.2.4 getId()

```
short Interval::getId ( )
```

Gets the id of the interval.

##### Returns

short Id of the interval

#### 6.2.2.5 getStartDate()

```
long long Interval::getStartDate ( )
```

Gets the start date of the interval.

##### Returns

long Start date of the interval

#### 6.2.2.6 getStopDate()

```
long long Interval::getStopDate ( )
```

Gets the stop date of the interval.

##### Returns

long Stop date of the interval

#### 6.2.2.7 getWeight()

```
short Interval::getWeight ( )
```

Gets the weight of the interval.

##### Returns

short Weight of the interval

#### 6.2.2.8 printInterval()

```
void Interval::printInterval ( )
```

Prints the interval.

#### 6.2.2.9 setStartDate()

```
void Interval::setStartDate (
    long long startDate )
```

Gets the point list of the interval.

##### Returns

vector<short> [Point](#) list of the interval

Sets the start date of the interval

##### Parameters

<i>startDate</i>	Start date of the interval
------------------	----------------------------

#### 6.2.2.10 setStopDate()

```
void Interval::setStopDate (
    long long stopDate )
```

Sets the stop date of the interval.

##### Parameters

<i>stopDate</i>	Stop date of the interval
-----------------	---------------------------

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

- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[Interval.h](#)
- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[Interval.cpp](#)

## 6.3 IntervalParser Class Reference

```
#include <IntervalParser.h>
```

## Public Member Functions

- [IntervalParser](#) ()  
*Construct a new [Interval](#) Parser:: [Interval](#) Parser object.*
- [~IntervalParser](#) ()  
*Destroy the [Interval](#) Parser:: [Interval](#) Parser object.*
- vector< [Interval](#) > [parse](#) (string fileName)  
*Parses the file and returns a vector of intervals.*
- void [sortIntervals](#) (vector< [Interval](#) > &intervalList)  
*Sorts the intervals by start date.*
- void [originToZero](#) (vector< [Interval](#) > &intervalList)  
*Sets the origin of the start date to 0.*

## 6.3.1 Constructor & Destructor Documentation

### 6.3.1.1 IntervalParser()

```
IntervalParser::IntervalParser ( )
```

Construct a new [Interval](#) Parser:: [Interval](#) Parser object.

### 6.3.1.2 ~IntervalParser()

```
IntervalParser::~~IntervalParser ( )
```

Destroy the [Interval](#) Parser:: [Interval](#) Parser object.

## 6.3.2 Member Function Documentation

### 6.3.2.1 originToZero()

```
void IntervalParser::originToZero (
    vector< Interval > & intervalList )
```

Sets the origin of the start date to 0.

#### Parameters

<i>intervalList</i>	The list of intervals to modify
---------------------	---------------------------------

### 6.3.2.2 parse()

```
vector< Interval > IntervalParser::parse (
    string fileName )
```

Parses the file and returns a vector of intervals.

#### Parameters

<i>fileName</i>	The PATH of the file to parse
-----------------	-------------------------------

#### Returns

vector<Interval>

### 6.3.2.3 sortIntervals()

```
void IntervalParser::sortIntervals (
    vector< Interval > & intervalList )
```

Sorts the intervals by start date.

#### Parameters

<i>intervalList</i>	The list of intervals to sort
---------------------	-------------------------------

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

- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[IntervalParser.h](#)
- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[IntervalParser.cpp](#)

## 6.4 Point Class Reference

```
#include <Point.h>
```

### Public Member Functions

- [Point](#) ()  
*Construct a new [Point::Point](#) object.*
- [Point](#) (short id)  
*Construct a new [Point::Point](#) object.*

- short [getId](#) ()  
*Gets the id of the point.*
- short [getCost](#) ()  
*Gets the cost of the point.*
- short [getMinInterval](#) ()  
*Gets the min interval of the point.*
- void [setCost](#) (short cost)  
*Sets the cost of the point.*
- void [setMinInterval](#) (short minInterval)  
*Sets the min interval of the point.*

## 6.4.1 Constructor & Destructor Documentation

### 6.4.1.1 [Point\(\)](#) [1/2]

```
Point::Point ( )
```

Construct a new [Point:: Point](#) object.

### 6.4.1.2 [Point\(\)](#) [2/2]

```
Point::Point (
    short id )
```

Construct a new [Point:: Point](#) object.

#### Parameters

<i>id</i>	<a href="#">Point</a> id
<i>cost</i>	<a href="#">Point</a> cost

## 6.4.2 Member Function Documentation

### 6.4.2.1 [getCost\(\)](#)

```
short Point::getCost ( )
```

Gets the cost of the point.

#### Returns

short Cost of the point

#### 6.4.2.2 getId()

```
short Point::getId ( )
```

Gets the id of the point.

##### Returns

short Id of the point

#### 6.4.2.3 getMinInterval()

```
short Point::getMinInterval ( )
```

Gets the min interval of the point.

##### Returns

short Min interval of the point

#### 6.4.2.4 setCost()

```
void Point::setCost (
    short cost )
```

Sets the cost of the point.

##### Parameters

<i>cost</i>	Cost of the point
-------------	-------------------

#### 6.4.2.5 setMinInterval()

```
void Point::setMinInterval (
    short minInterval )
```

Sets the min interval of the point.

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

- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[Point.h](#)
- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[Point.cpp](#)

## 6.5 SolutionParser Class Reference

```
#include <SolutionParser.h>
```

### Public Member Functions

- [SolutionParser](#) ()
- [~SolutionParser](#) ()
- void [writeCSV](#) (vector< [Interval](#) \* > solution)

### 6.5.1 Constructor & Destructor Documentation

#### 6.5.1.1 SolutionParser()

```
SolutionParser::SolutionParser ( )
```

#### 6.5.1.2 ~SolutionParser()

```
SolutionParser::~~SolutionParser ( )
```

### 6.5.2 Member Function Documentation

#### 6.5.2.1 writeCSV()

```
void SolutionParser::writeCSV (
    vector< Interval * > solution )
```

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

- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[SolutionParser.h](#)
- E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/[SolutionParser.cpp](#)





## Chapter 7

# File Documentation

### 7.1 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ Greedy.cpp File Reference

```
#include "Greedy.h"
```

### 7.2 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ Greedy.h File Reference

```
#include "Interval.h"  
#include "Point.h"  
#include <vector>  
#include <algorithm>
```

#### Classes

- class [Greedy](#)

### 7.3 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ Greedy.h

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

```
00001 #pragma once  
00002 #include "Interval.h"  
00003 #include "Point.h"  
00004 #include <vector>  
00005 #include <algorithm>  
00006  
00007 using namespace std;  
00008  
00009 class Greedy{  
00010 private:  
00011     vector<Interval> omega;  
00012     vector<Interval *> solution;
```

```

00013     vector<Interval *> qi;
00014     vector<Interval *> ci;
00015     Interval * iPrime;
00016     vector<Point> points;
00017     vector<short> first;
00018     vector<short> last;
00019     short maxChannel;
00020     float alpha;
00022 public:
00023     Greedy();
00024     Greedy(vector<Interval> omega, short maxChannel, float alpha);
00025
00026     void setOmega(vector<Interval> omega);
00027     void setMaxChannel(short maxChannel);
00028     void setAlpha(float alpha);
00029
00030     void findQi(int omega_id);
00031     void findPoints();
00032     void findCi();
00033     bool exchangeCheck(Interval * i);
00034
00035     vector<Interval *> solve();
00036 };

```

## 7.4 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Interval.cpp File Reference

```
#include "Interval.h"
```

## 7.5 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Interval.h File Reference

```
#include <iostream>
#include <vector>
```

### Classes

- class [Interval](#)

## 7.6 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Interval.h

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

```

00001 #pragma once
00002 #include <iostream>
00003 #include <vector>
00004
00005 using namespace std;
00006 class Interval{
00007 private:
00008     short id;
00009     long long startDate;
00010     long long stopDate;
00011     string channel;
00012     string hashcode;
00013     short weight;
00014     //vector<short> pointList;
00015

```

```

00016 public :
00017     Interval();
00018     Interval(short id, long long startDate, long long stopDate, string channel, string hashCode);
00019
00020     short getId();
00021     long long getStartDate();
00022     long long getStopDate();
00023     string getChannel();
00024     string getHashCode();
00025     short getWeight();
00026     //vector<short> getPointList();
00027     void setStartDate(long long startDate);
00028     void setStopDate(long long stopDate);
00029     //void setPointList(vector<short> pointList);
00030
00031     //void addPoint(short point);
00032     //short first();
00033     //short last();
00034     void generateRandomWeight();
00035     void printInterval();
00036
00037 };
00038

```

## 7.7 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/IntervalParser.cpp File Reference

```
#include "IntervalParser.h"
```

## 7.8 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/IntervalParser.h File Reference

```

#include "Interval.h"
#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>
#include <string>
#include <algorithm>

```

### Classes

- class [IntervalParser](#)

### Macros

- #define [IntervalParser8H](#)

### 7.8.1 Macro Definition Documentation

### 7.8.1.1 IntervalParser8H

```
#define IntervalParser8H
```

## 7.9 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ IntervalParser.h

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

```
00001 #pragma once
00002
00003 #ifndef IntervalParser_H
00004 #define IntervalParser8H
00005
00006 #include "Interval.h"
00007 #include <iostream>
00008 #include <fstream>
00009 #include <vector>
00010 #include <sstream>
00011 #include <string>
00012 #include <algorithm>
00013
00014 using namespace std;
00015
00016 class IntervalParser{
00017 private:
00018     /* data */
00019 public:
00020     IntervalParser();
00021     ~IntervalParser();
00022
00023     vector<Interval> parse(string fileName);
00024     void sortIntervals(vector<Interval> &intervalList);
00025     void originToZero(vector<Interval> &intervalList);
00026 };
00027
00028 #endif
```

## 7.10 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/↵ Greedy/main.cpp File Reference

```
#include <iostream>
#include "interval.h"
#include "intervalParser.h"
#include "solutionParser.h"
#include "greedy.h"
#include "Point.h"
```

### Functions

- int [main](#) ()

### 7.10.1 Function Documentation

### 7.10.1.1 main()

```
int main ( )
```

## 7.11 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ Point.cpp File Reference

```
#include "Point.h"
```

## 7.12 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ Point.h File Reference

### Classes

- class [Point](#)

## 7.13 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ Point.h

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

```
00001 #pragma once
00002
00003 class Point{
00004 private :
00005     short id;
00006     short cost;
00007     short minInterval; //TODO: rename
00008
00009 public :
00010     Point();
00011     Point(short id);
00012     short getId();
00013     short getCost();
00014     short getMinInterval();
00015     void setCost(short cost);
00016     void setMinInterval(short minInterval);
00017 };
00018
```

## 7.14 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ Python/1\_generate\_latency.py File Reference

### Namespaces

- namespace [1\\_generate\\_latency](#)

## Functions

- def [1\\_generate\\_latency.nomarlizeText](#) (element)
- def [1\\_generate\\_latency.check\\_timestamp\\_valid](#) (filename\_str, date\_valid)
- def [1\\_generate\\_latency.convert\\_xlm\\_file\\_to\\_date](#) (filename\_str)
- def [1\\_generate\\_latency.checking](#) ()
- def [1\\_generate\\_latency.read\\_xml](#) (base\_path, input\_file, channel\_c)
- def [1\\_generate\\_latency.main](#) ()

## Variables

- list [1\\_generate\\_latency.data\\_list](#) = []

## 7.15 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Python/common.py File Reference ↩

### Namespaces

- namespace [common](#)

### Functions

- def [common.get\\_channel\\_name\\_by\\_code](#) (channel\_code)
- def [common.check\\_file](#) (file\_name, path\_name)
- def [common.map\\_channel](#) (channel\_code)

## 7.16 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Python/README.md File Reference ↩

## 7.17 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/SolutionParser.cpp File Reference ↩

```
#include "SolutionParser.h"
```

## 7.18 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/SolutionParser.h File Reference ↩

```
#include <iostream>
#include <fstream>
#include <vector>
#include <sstream>
#include "Interval.h"
```

## Classes

- class [SolutionParser](#)

## 7.19 E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/↵ SolutionParser.h

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

```
00001 #pragma once
00002 #include <iostream>
00003 #include <fstream>
00004 #include <vector>
00005 #include <sstream>
00006 #include "Interval.h"
00007
00008 class SolutionParser{
00009 private:
00010     /* data */
00011 public:
00012     SolutionParser(/* args */);
00013     ~SolutionParser();
00014
00015     void writeCSV(vector<Interval *> solution);
00016 };
00017
00018
```





# Index

~IntervalParser  
  IntervalParser, 21

~SolutionParser  
  SolutionParser, 25

1\_generate\_latency, 9  
  check\_timestamp\_valid, 9  
  checking, 9  
  convert\_xlm\_file\_to\_date, 9  
  data\_list, 10  
  main, 10  
  nomarlizeText, 10  
  read\_xml, 10

check\_file  
  common, 11

check\_timestamp\_valid  
  1\_generate\_latency, 9

checking  
  1\_generate\_latency, 9

common, 10  
  check\_file, 11  
  get\_channel\_name\_by\_code, 11  
  map\_channel, 11

convert\_xlm\_file\_to\_date  
  1\_generate\_latency, 9

data\_list  
  1\_generate\_latency, 10

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Python/convert\_xlm\_file\_to\_date.py, 27

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/Interval.h, 27

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/Interval.cpp, 28

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/Interval.h, 28

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/IntervalParser.cpp, 29

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/IntervalParser.h, 29

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/main.cpp, 30

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/main.h, 31

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/Point.h, 31

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/Python/1\_generate\_latency.py, 31

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Greedy/Python/convert\_xlm\_file\_to\_date.py, 32

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Python/Interval.cpp, 32

E:/Poly/DI5/PRD/Algo\_workspace/GreedyAlpha/Greedy/Greedy/Python/Interval.h, 32

exchangeCheck  
  Greedy, 14

findCi  
  Greedy, 14

findPoints  
  Greedy, 15

findQi  
  Greedy, 15

generateRandomWeight  
  Interval, 17

get\_channel\_name\_by\_code  
  common, 11

getChannel  
  Interval, 18

getCost  
  Point, 23

getHashcode  
  Interval, 18

getId  
  Interval, 18

Interval.h, 27

Interval.cpp, 28

Interval.h, 28

IntervalParser.cpp, 29

IntervalParser.h, 29

main.cpp, 30

main.h, 31

Point.h, 31

Python/1\_generate\_latency.py, 31

Python/convert\_xlm\_file\_to\_date.py, 32

Python/Interval.cpp, 32

Python/Interval.h, 32

exchangeCheck, 14

findPoints, 15

findQi, 15

Greedy, 13, 14

setMaxChannel, 15

solve, 16

- Interval, 16
  - generateRandomWeight, 17
  - getChannel, 18
  - getHashCode, 18
  - getId, 18
  - getStartDate, 19
  - getStopDate, 19
  - getWeight, 19
  - Interval, 17
  - printInterval, 19
  - setStartDate, 20
  - setStopDate, 20
- IntervalParser, 20
  - ~IntervalParser, 21
  - IntervalParser, 21
  - originToZero, 21
  - parse, 22
  - sortIntervals, 22
- IntervalParser.h
  - IntervalParser8H, 29
- IntervalParser8H
  - IntervalParser.h, 29
- main
  - 1\_generate\_latency, 10
  - main.cpp, 30
- main.cpp
  - main, 30
- map\_channel
  - common, 11
- nomarlizeText
  - 1\_generate\_latency, 10
- originToZero
  - IntervalParser, 21
- parse
  - IntervalParser, 22
- Point, 22
  - getCost, 23
  - getId, 23
  - getMinInterval, 24
  - Point, 23
  - setCost, 24
  - setMinInterval, 24
- printInterval
  - Interval, 19
- read\_xml
  - 1\_generate\_latency, 10
- setAlpha
  - Greedy, 15
- setCost
  - Point, 24
- setMaxChannel
  - Greedy, 15
- setMinInterval
  - Point, 24
- setOmega
  - Greedy, 16
- setStartDate
  - Interval, 20
- setStopDate
  - Interval, 20
- SolutionParser, 25
  - ~SolutionParser, 25
  - SolutionParser, 25
  - writeCSV, 25
- solve
  - Greedy, 16
- sortIntervals
  - IntervalParser, 22
- writeCSV
  - SolutionParser, 25