Greedy

Generated by Doxygen 1.9.6

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/1wvL6HJwhUpC1QFqRUHy1WuobupTNrCR1?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.

Sepcifically, 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:

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 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 1_generate_latency.convert_xlm_file_to_date (
              filename_str )
```

5.1.1.4 main()

```
def 1_generate_latency.main ( )
```

5.1.1.5 nomarlizeText()

```
def 1_generate_latency.nomarlizeText (
              element )
```

5.1.1.6 read_xml()

```
def 1_generate_latency.read_xml (
              base_path,
              input_file,
              channel_c )
```

5.1.2 Variable Documentation

5.1.2.1 data_list

```
list 1_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

omega	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

point↵ Id	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

omega↵ _id	Id of the interval in the omega list

6.1.2.5 setAlpha()

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

Sets a new alpha value.

Parameters

alpha	

6.1.2.6 setMaxChannel()

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

Sets a new max channel.

Parameters

maxChannel	

6.1.2.7 setOmega()

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

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

Parameters

omega	

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

id	Interval id
startDate	Interval start date
stopDate	Interval stop date
channel	Interval channel
hashcode	Interval hashcode

6.2.2 Member Function Documentation

6.2.2.1 generateRandomWeight()

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

Sets the point list of the interval.

Parameters

pointList	Point list of the interval

Adds a point to the point list of the interval

Parameters

point	Point 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

startDate	Start date of the interval

6.2.2.10 setStopDate()

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

Sets the stop date of the interval.

Parameters

stopDate	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

intervalList	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

fileName	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

intervalList	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

id	Point id
cost	Point 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

cost	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.

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

using namespace std;

class Greedy{
private:
    vector<Interval> omega;
    vector<Interval *> solution;
    vector<Interval *> qi;
    vector<Interval *> ci;
    Interval * iPrime;
    vector<Point> points;
    vector<short> first;
    vector<short> last;
    short maxChannel;
    float alpha;
public:
    Greedy();
    Greedy(vector<Interval> omega, short maxChannel, float alpha);

    void setOmega(vector<Interval> omega);
    void setMaxChannel(short maxChannel);
    void setAlpha(float alpha);

    void findQi(int omega_id);
    void findPoints();
    void findCi();
    bool exchangeCheck(Interval * i);

    vector<Interval *> solve();
};
```

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.

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

using namespace std;
class Interval{
private:
    short id;
    long long startDate;
    long long stopDate;
    string channel;
    string hashcode;
    short weight;
    //vector<short> pointList;

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

    short getId();
    long long getStartDate();
    long long getStopDate();
    string getChannel();
    string getHashcode();
    short getWeight();
    //vector<short> getPointList();
    void setStartDate(long long startDate);
    void setStopDate(long long stopDate);
    //void setPointList(vector<short> pointList);

    //void addPoint(short point);
    //short first();
    //short last();
    void generateRandomWeight();
    void printInterval();

};

```

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.

```
#pragma once

#ifndef IntervalParser_H
#define IntervalParser8H

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

using namespace std;

class IntervalParser{
private:
    /* data */
public:
    IntervalParser();
    ~IntervalParser();

    vector<Interval> parse(string fileName);
    void sortIntervals(vector<Interval> &intervalList);
    void originToZero(vector<Interval> &intervalList);
};

#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.

```
#pragma once

class Point{
private :
    short id;
    short cost;
    short minInterval; //TODO: rename

public :
    Point();
    Point(short id);
    short getId();
    short getCost();
    short getMinInterval();
    void setCost(short cost);
    void setMinInterval(short minInterval);
};
```

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.

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

class SolutionParser{
private:
    /* data */
public:
    SolutionParser(/* args */);
    ~SolutionParser();

    void writeCSV(vector<Interval *> solution);
};


```

Index