ExpositoTOP

2

Generated by Doxygen 1.8.20

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

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 Package es

Packages

- package ull

5.2 Package es.ull

Packages

- package esit

5.3 Package es.ull.esit

Packages

- package utilities
- package utils

5.4 Package es.ull.esit.utilities

Classes

- class BellmanFord

  Implementation of BellmanFord algorithm.
- class ExpositoUtilities

  Different auxiliary utilities to be used along the project.
- class PowerSet

  Class to calculate every subset of a given set.

5.5 Package es.ull.esit.utils

Classes

- class Pair

  Class to represent a generic pair of objects.

5.6 Package top

Classes

- class mainTOPTW

  Main program class.
- class TOPTW
- class TOPTWEvaluator
- class TOPTWGRASP
- class TOPTWReader

  Class read a TOPTW problem.
- class TOPTWRoute

  Class to represent the route.
- class TOPTWSolution

  Class to represent the TOPTW problem solution.

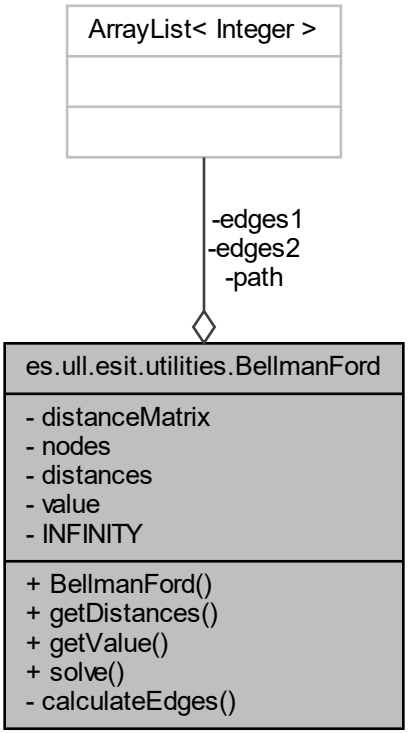
Chapter 6

Class Documentation

6.1 es.ull.esit.utilities.BellmanFord Class Reference

Implementation of BellmanFord algorithm.

Collaboration diagram for es.ull.esit.utilities.BellmanFord:



Public Member Functions

- BellmanFord (int[ ][ ] distanceMatrix, int nodes, ArrayList< Integer > path)

  Class constructor.
- int[ ] getDistances ()

  Getter for distances array.
- int getValue ()

  Getter for path cost.
- void solve ()

  Method for solving the stored problem.

Private Member Functions

- void calculateEdges ()

  Auxiliary method to calculate edges.

Private Attributes

- final int[ ][ ] distanceMatrix
- ArrayList< Integer > edges1 = null
- ArrayList< Integer > edges2 = null
- final int nodes
- final ArrayList< Integer > path
- int[ ] distances = null
- int value

Static Private Attributes

- static final int INFINITY = 999999

6.1.1 Detailed Description

Implementation of BellmanFord algorithm.

This class implements the BellmandFord search algorithm using integer distances

6.1.2 Constructor & Destructor Documentation

6.1.2.1 BellmanFord()

```
es.ull.esit.utilities.BellmanFord.BellmanFord (
            int distanceMatrix[][],
            int nodes,
            ArrayList< Integer > path )
```

Class constructor.

Parameters

distanceMatrix	-> Matrix to store distances
nodes	-> number of nodes
path	-> final path

6.1.3 Member Function Documentation

6.1.3.1 calculateEdges()

```
void es.ull.esit.utilities.BellmanFord.calculateEdges ( )  [private]
```

Auxiliary method to calculate edges.

6.1.3.2 getDistances()

```
int [] es.ull.esit.utilities.BellmanFord.getDistances ( )
```

Getter for distances array.

Returns

int[] -> unidimensional array of distances

6.1.3.3 getValue()

```
int es.ull.esit.utilities.BellmanFord.getValue ( )
```

Getter for path cost.

Returns

int -> Cost of the optimal path found

6.1.3.4 solve()

```
void es.ull.esit.utilities.BellmanFord.solve ( )
```

Method for solving the stored problem.

6.1.4 Member Data Documentation

6.1.4.1 distanceMatrix

```
final int [][] es.ull.esit.utilities.BellmanFord.distanceMatrix  [private]
```

Matrix to store distances.

6.1.4.2 distances

```
int [] es.ull.esit.utilities.BellmanFord.distances = null  [private]
```

Auxiliary distance array.

6.1.4.3 edges1

```
ArrayList<Integer> es.ull.esit.utilities.BellmanFord.edges1 = null  [private]
```

Right to left edges.

6.1.4.4 edges2

```
ArrayList<Integer> es.ull.esit.utilities.BellmanFord.edges2 = null  [private]
```

Left to right edges.

6.1.4.5 INFINITY

```
final int es.ull.esit.utilities.BellmanFord.INFINITY = 999999  [static], [private]
```

Infinity constant.

6.1.4.6 nodes

```
final int es.ull.esit.utilities.BellmanFord.nodes  [private]
```

Number of nodes.

6.1.4.7 path

```
final ArrayList<Integer> es.ull.esit.utilities.BellmanFord.path  [private]
```

Final path.

6.1.4.8 value

```
int es.ull.esit.utilities.BellmanFord.value  [private]
```

Path cost.

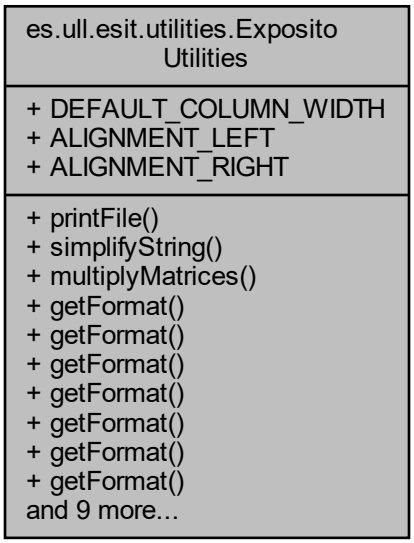
The documentation for this class was generated from the following file:

- src/main/java/es/ull/esit/utilities/BellmanFord.java

6.2 es.ull.esit.utilities.ExpositoUtilities Class Reference

Different auxiliary utilities to be used along the project.

Collaboration diagram for es.ull.esit.utilities.ExpositoUtilities:



Static Public Member Functions

- static void printFile (String file)

  Method for printing files.
- static String simplifyString (String string)

  Parser to simplify strings containing undesirable characters.
- static double[ ][ ] multiplyMatrices (double[ ][ ] a, double[ ][ ] b)

  Method to multiply 2 double matrix.
- static String getFormat (String string)

  Method to get the format of a given string.
- static String getFormat (double value)

Double to string formatter.

- static String getFormat (double value, int zeros)

  Double to string formatter.
- static String getFormat (String string, int width)

  Auxiliary method for getting a string format.
- static String getFormat (String string, int width, int alignment)

  Auxiliary method for getting a string format.
- static String getFormat (ArrayList< String > strings, int width)
- static String getFormat (ArrayList< Integer > strings)
- static String getFormat (String[ ] strings, int width)
- static String getFormat (String[ ][ ] matrixStrings, int width)
- static String getFormat (String[ ] strings)
- static String getFormat (String[ ] strings, int[ ] width)
- static String getFormat (String[ ] strings, int[ ] width, int[ ] alignment)
- static boolean isInteger (String str)

  Check is a given number is integer.
- static boolean isDouble (String str)

  Check is a given number is double.
- static boolean isAcyclic (int[ ][ ] distanceMatrix)

  Check is a graph is acyclic.
- static boolean thereIsPath (int[ ][ ] distanceMatrix, int node)

  Checks if a given node is reachable.

Static Public Attributes

- static final int DEFAULT_COLUMN_WIDTH = 10
- static final int ALIGNMENT_LEFT = 1
- static final int ALIGNMENT_RIGHT = 2

6.2.1 Detailed Description

Different auxiliary utilities to be used along the project.

This class implements a number of methods which will be used in the project. It will be used as a library.

6.2.2 Member Function Documentation

6.2.2.1 getFormat() [1/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            ArrayList< Integer > strings )  [static]
```

Parameters

strings	-> strings to be analyzed

Returns

String -> formatted string

6.2.2.2 getFormat() [2/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            ArrayList< String > strings,
            int width )  [static]
```

Parameters

strings	-> strings to be analyzed
width	-> width of the string

Returns

String -> string format

6.2.2.3 getFormat() [3/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            double value )  [static]
```

Double to string formatter.

Parameters

value	-> value to be formatted

Returns

String -> formatted result

6.2.2.4 getFormat() [4/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            double value,
            int zeros )  [static]
```

Double to string formatter.

Parameters

value	-> value to be formatted
zeros	-> desired decimal precision

Returns

String -> formatted result

6.2.2.5 getFormat() [5/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            String string )  [static]
```

Method to get the format of a given string.

Parameters

string	-> string to be analyzed

Returns

String -> result of the analysis

6.2.2.6 getFormat() [6/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            String string,
            int width )  [static]
```

Auxiliary method for getting a string format.

Parameters

string	-> string to be analyzed
width	-> width of the string

Returns

String -> string format

6.2.2.7 getFormat() [7/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            String string,
            int width,
            int alignment )  [static]
```

Auxiliary method for getting a string format.

Parameters

string	-> string to be analyzed
width	-> width of the string
alignment	-> string alignment

Returns

String -> string format

6.2.2.8 getFormat() [8/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            String[] strings )  [static]
```

Parameters

strings	-> String of strings to be analyzed

Returns

String -> string format

6.2.2.9 getFormat() [9/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            String[] strings,
            int width )  [static]
```

Parameters

strings	-> strings to be analyzed
width	-> string width

Returns

String -> string format

6.2.2.10 getFormat() [10/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            String[] strings,
            int[] width )  [static]
```

Parameters

strings	-> String of strings to be analyzed
width	-> string width

Returns

String -> string format

6.2.2.11 getFormat() [11/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            String[] strings,
            int[] width,
            int[] alignment )  [static]
```

Parameters

strings	-> String of strings to be analyzed
width	-> string width
alignment	-> string alignment

Returns

String -> string foramt

6.2.2.12 getFormat() [12/12]

```
static String es.ull.esit.utilities.ExpositoUtilities.getFormat (
            String matrixStrings[][],
            int width )  [static]
```

6.2.2.13 isAcyclic()

```
static boolean es.ull.esit.utilities.ExpositoUtilities.isAcyclic (
            int distanceMatrix[][] )  [static]
```

Check is a graph is acyclic.

Parameters

distanceMatrix	-> Matrix distances

Returns

boolean -> True or false

6.2.2.14 isDouble()

```
static boolean es.ull.esit.utilities.ExpositoUtilities.isDouble (
            String str )  [static]
```

Check is a given number is double.

Parameters

str	-> string containing the number

Returns

boolean -> True of false

6.2.2.15 isInteger()

```
static boolean es.ull.esit.utilities.ExpositoUtilities.isInteger (
            String str )  [static]
```

Check is a given number is integer.

Parameters

str	-> string containing the number

Returns

boolean -> True of false

6.2.2.16 multiplyMatrices()

```
static double [][] es.ull.esit.utilities.ExpositoUtilities.multiplyMatrices (
            double a[][],
            double b[][] )  [static]
```

Method to multiply 2 double matrix.

Parameters

a	-> Left matrix
b	-> Right matrix

Returns

double[][] -> Matrix product result

6.2.2.17 printFile()

```
static void es.ull.esit.utilities.ExpositoUtilities.printFile (
            String file )  [static]
```

Method for printing files.

Parameters

file	-> filename

6.2.2.18 simplifyString()

```
static String es.ull.esit.utilities.ExpositoUtilities.simplifyString (
            String string )  [static]
```

Parser to simplify strings containing undesirable characters.

Parameters

string	-> String to be simplified

Returns

String -> simplified string

6.2.2.19 thereIsPath()

```
static boolean es.ull.esit.utilities.ExpositoUtilities.thereIsPath (
            int distanceMatrix[][],
            int node )  [static]
```

Checks if a given node is reachable.

Parameters

distanceMatrix	-> Matrix distances
node	-> goal node

Returns

boolean -> True or false

6.2.3 Member Data Documentation

6.2.3.1 ALIGNMENT_LEFT

```
final int es.ull.esit.utilities.ExpositoUtilities.ALIGNMENT_LEFT = 1  [static]
```

Constant to define left alignment

6.2.3.2 ALIGNMENT_RIGHT

```
final int es.ull.esit.utilities.ExpositoUtilities.ALIGNMENT_RIGHT = 2  [static]
```

Constant to define right alignment

6.2.3.3 DEFAULT_COLUMN_WIDTH

```
final int es.ull.esit.utilities.ExpositoUtilities.DEFAULT_COLUMN_WIDTH = 10  [static]
```

Constant to define column width

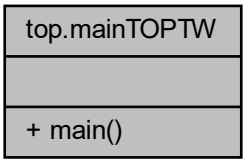
The documentation for this class was generated from the following file:

- src/main/java/es/ull/esit/utilities/ExpositoUtilities.java

6.3 top.mainTOPTW Class Reference

Main program class.

Collaboration diagram for top.mainTOPTW:



Static Public Member Functions

- static void main (String[ ] args)

6.3.1 Detailed Description

Main program class.

This class loads the distance graphs contained in txt files and calculates the optimal path.

6.3.2 Member Function Documentation

6.3.2.1 main()

```
static void top.mainTOPTW.main (
            String[] args )  [static]
```

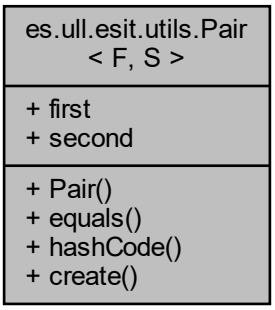
The documentation for this class was generated from the following file:

- src/main/java/top/mainTOPTW.java

6.4 es.ull.esit.utils.Pair< F, S > Class Template Reference

Class to represent a generic pair of objects.

Collaboration diagram for es.ull.esit.utils.Pair< F, S >:



Public Member Functions

- Pair (F first, S second)
  Constructor.
- boolean equals (Object o)
  Check is a pair is equal to another.
- int hashCode ()
  HashCode of the pair.

Static Public Member Functions

- static< A, B > Pair< A, B > create (A a, B b)
  Creates a new pair.

Public Attributes

- final F first
- final S second

6.4.1 Detailed Description

Class to represent a generic pair of objects.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Pair()

```
es.ull.esit.utils.Pair< F, S >.Pair (
            F first,
            S second )
```

Constructor.

Parameters

first	-> first pair value
second	-> second pair value

6.4.3 Member Function Documentation

6.4.3.1 create()

```
static <A, B> Pair<A, B> es.ull.esit.utils.Pair< F, S >.create (
            A a,
            B b )  [static]
```

Creates a new pair.

Parameters

a	-> first pair value
b	-> second pair value

Returns

Pair -> created pair

6.4.3.2 equals()

```
boolean es.ull.esit.utils.Pair< F, S >.equals (
            Object o )
```

Check is a pair is equal to another.

Parameters

o	-> comparison

Returns

boolean -> True or false

6.4.3.3 hashCode()

```
int es.ull.esit.utils.Pair< F, S >.hashCode ( )
```

HashCode of the pair.

Returns

int -> hashCode

6.4.4 Member Data Documentation

6.4.4.1 first

```
final F es.ull.esit.utils.Pair< F, S >.first
```

First pair value.

6.4.4.2 second

```
final S es.ull.esit.utils.Pair< F, S >.second
```

Second pair value.

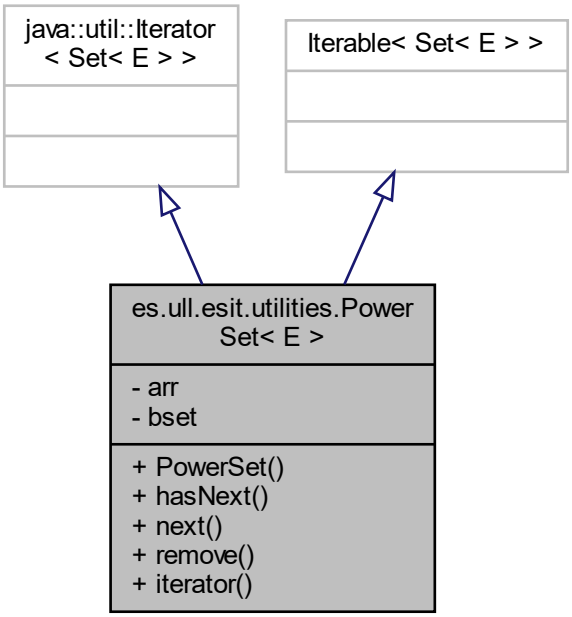
The documentation for this class was generated from the following file:

- src/main/java/es/ull/esit/utils/Pair.java

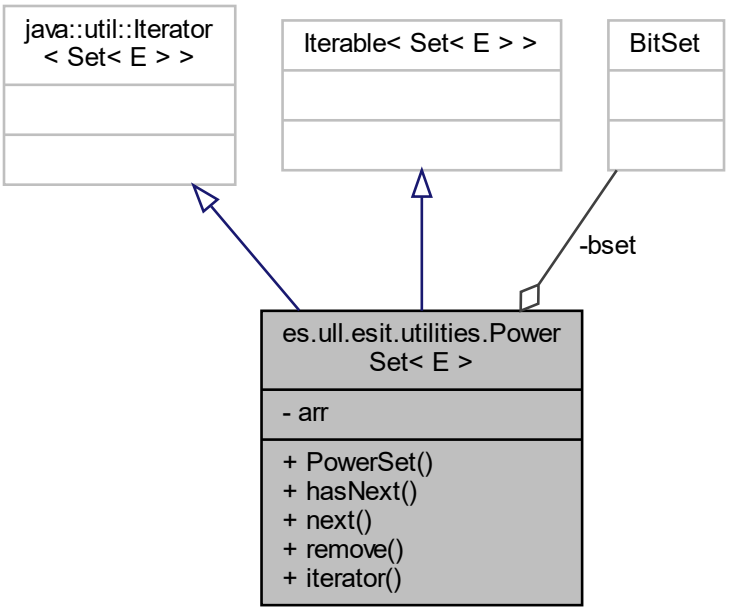
6.5 es.ull.esit.utilities.PowerSet< E > Class Template Reference

Class to calculate every subset of a given set.

Inheritance diagram for es.ull.esit.utilities.PowerSet< E >:



Collaboration diagram for es.ull.esit.utilities.PowerSet< E >:



Public Member Functions

- PowerSet (Set< E > set)

  Class constructor.
- boolean hasNext ()

  Check if a subset has a next subset.
- Set< E > next ()

  Calculate next subset.
- void remove ()
- Iterator< Set< E > > iterator ()

Private Attributes

- final E[ ] arr
- final BitSet bset

6.5.1 Detailed Description

Class to calculate every subset of a given set.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 PowerSet()

```
es.ull.esit.utilities.PowerSet< E >.PowerSet (
            Set< E > set )
```

Class constructor.

Parameters

set	-> set to calculate its subsets

6.5.3 Member Function Documentation

6.5.3.1 hasNext()

```
boolean es.ull.esit.utilities.PowerSet< E >.hasNext ( )
```

Check if a subset has a next subset.

Returns

boolean -> True or false

6.5.3.2 iterator()

```
Iterator<Set<E> > es.ull.esit.utilities.PowerSet< E >.iterator ( )
```

6.5.3.3 next()

```
Set<E> es.ull.esit.utilities.PowerSet< E >.next ( )
```

Calculate next subset.

Returns

Set<E> -> set result

6.5.3.4 remove()

```
void es.ull.esit.utilities.PowerSet< E >.remove ( )
```

6.5.4 Member Data Documentation

6.5.4.1 arr

```
final E [] es.ull.esit.utilities.PowerSet< E >.arr  [private]
```

Array set.

6.5.4.2 bset

```
final BitSet es.ull.esit.utilities.PowerSet< E >.bset  [private]
```

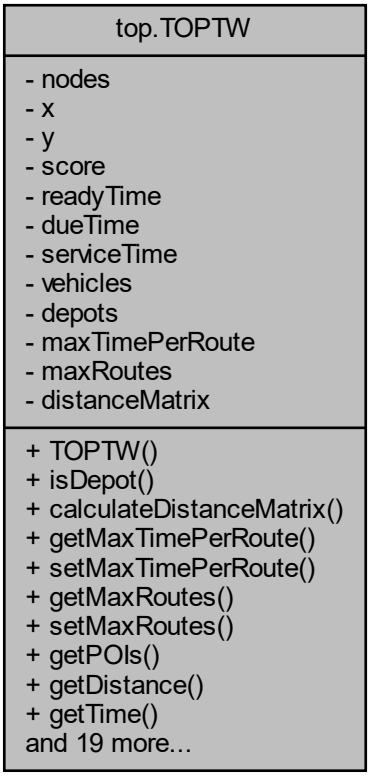
Bitset.

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

- src/main/java/es/ull/esit/utilities/PowerSet.java

6.6 top.TOPTW Class Reference

Collaboration diagram for top.TOPTW:



Public Member Functions

- TOPTW (int nodes, int routes)
- boolean isDepot (int a)
- void calculateDistanceMatrix ()
- double getMaxTimePerRoute ()
- void setMaxTimePerRoute (double maxTimePerRoute)
- double getMaxRoutes ()
- void setMaxRoutes (double maxRoutes)
- int getPOIs ()
- double getDistance (int i, int j)
- double getTime (int i, int j)
- int getNodes ()
- void setNodes (int nodes)
- double getX (int index)
- void setX (int index, double x)
- double getY (int index)
- void setY (int index, double y)
- double getScore (int index)
- double[ ] getScore ()
- void setScore (int index, double score)
- double getReadyTime (int index)
- void setReadyTime (int index, double readyTime)
- double getDueTime (int index)
- void setDueTime (int index, double dueTime)
- double getServiceTime (int index)
- void setServiceTime (int index, double serviceTime)
- int getVehicles ()
- String toString ()
- int addNode ()
- int addNodeDepot ()

Private Attributes

- int nodes
- final double[ ] x
- final double[ ] y
- final double[ ] score
- final double[ ] readyTime
- final double[ ] dueTime
- final double[ ] serviceTime
- final int vehicles
- int depots
- double maxTimePerRoute
- double maxRoutes
- final double[ ][ ] distanceMatrix

6.6.1 Constructor & Destructor Documentation

6.6.1.1 TOPTW()

```
top.TOPTW.TOPTW (
            int nodes,
            int routes )
```

6.6.2 Member Function Documentation

6.6.2.1 addNode()

```
int top.TOPTW.addNode ( )
```

6.6.2.2 addNodeDepot()

```
int top.TOPTW.addNodeDepot ( )
```

6.6.2.3 calculateDistanceMatrix()

```
void top.TOPTW.calculateDistanceMatrix ( )
```

6.6.2.4 getDistance()

```
double top.TOPTW.getDistance (
            int i,
            int j )
```

6.6.2.5 getDueTime()

```
double top.TOPTW.getDueTime (
            int index )
```

6.6.2.6 getMaxRoutes()

```
double top.TOPTW.getMaxRoutes ( )
```

6.6.2.7 getMaxTimePerRoute()

```
double top.TOPTW.getMaxTimePerRoute ( )
```

6.6.2.8 getNodes()

```
int top.TOPTW.getNodes ( )
```

6.6.2.9 getPOIs()

```
int top.TOPTW.getPOIs ( )
```

6.6.2.10 getReadyTime()

```
double top.TOPTW.getReadyTime (
            int index )
```

6.6.2.11 getScore() [1/2]

```
double [] top.TOPTW.getScore ( )
```

6.6.2.12 getScore() [2/2]

```
double top.TOPTW.getScore (
            int index )
```

6.6.2.13 getServiceTime()

```
double top.TOPTW.getServiceTime (
            int index )
```

6.6.2.14 getTime()

```
double top.TOPTW.getTime (
            int i,
            int j )
```

6.6.2.15 getVehicles()

```
int top.TOPTW.getVehicles ( )
```

6.6.2.16 getX()

```
double top.TOPTW.getX (
            int index )
```

6.6.2.17 getY()

```
double top.TOPTW.getY (
            int index )
```

6.6.2.18 isDepot()

```
boolean top.TOPTW.isDepot (
            int a )
```

6.6.2.19 setDueTime()

```
void top.TOPTW.setDueTime (
            int index,
            double dueTime )
```

6.6.2.20 setMaxRoutes()

```
void top.TOPTW.setMaxRoutes (
            double maxRoutes )
```

6.6.2.21 setMaxTimePerRoute()

```
void top.TOPTW.setMaxTimePerRoute (
            double maxTimePerRoute )
```

6.6.2.22 setNodes()

```
void top.TOPTW.setNodes (
            int nodes )
```

6.6.2.23 setReadyTime()

```
void top.TOPTW.setReadyTime (
            int index,
            double readyTime )
```

6.6.2.24 setScore()

```
void top.TOPTW.setScore (
            int index,
            double score )
```

6.6.2.25 setServiceTime()

```
void top.TOPTW.setServiceTime (
            int index,
            double serviceTime )
```

6.6.2.26 setX()

```
void top.TOPTW.setX (
            int index,
            double x )
```

6.6.2.27 setY()

```
void top.TOPTW.setY (
            int index,
            double y )
```

6.6.2.28 toString()

```
String top.TOPTW.toString ( )
```

6.6.3 Member Data Documentation

6.6.3.1 depots

```
int top.TOPTW.depots  [private]
```

6.6.3.2 distanceMatrix

```
final double [][] top.TOPTW.distanceMatrix  [private]
```

6.6.3.3 dueTime

```
final double [] top.TOPTW.dueTime  [private]
```

6.6.3.4 maxRoutes

```
double top.TOPTW.maxRoutes  [private]
```

6.6.3.5 maxTimePerRoute

```
double top.TOPTW.maxTimePerRoute  [private]
```

6.6.3.6 nodes

```
int top.TOPTW.nodes  [private]
```

6.6.3.7 readyTime

```
final double [] top.TOPTW.readyTime  [private]
```

6.6.3.8 score

```
final double [] top.TOPTW.score  [private]
```

6.6.3.9 serviceTime

```
final double [] top.TOPTW.serviceTime  [private]
```

6.6.3.10 vehicles

```
final int top.TOPTW.vehicles  [private]
```

6.6.3.11 x

```
final double [] top.TOPTW.x  [private]
```

6.6.3.12 y

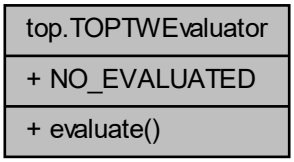
```
final double [] top.TOPTW.y  [private]
```

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

- src/main/java/top/TOPTW.java

6.7 top.TOPTWEvaluator Class Reference

Collaboration diagram for top.TOPTWEvaluator:



Public Member Functions

- void evaluate (TOPTWSolution solution)

Static Public Attributes

- static double NO_EVALUATED = -1.0

6.7.1 Member Function Documentation

6.7.1.1 evaluate()

```
void top.TOPTWEvaluator.evaluate (
            TOPTWSolution solution )
```

6.7.2 Member Data Documentation

6.7.2.1 NO_EVALUATED

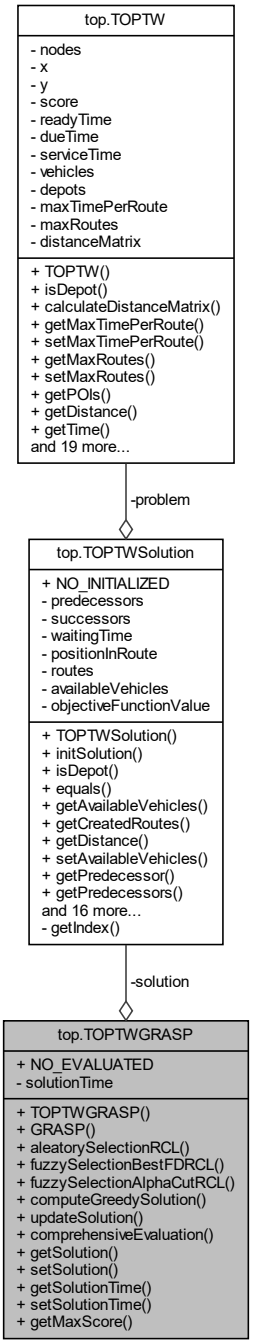
```
double top.TOPTWEvaluator.NO_EVALUATED = -1.0  [static]
```

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

- src/main/java/top/TOPTWEvaluator.java

6.8 top.TOPTWGRASP Class Reference

Collaboration diagram for top.TOPTWGRASP:



Public Member Functions

- TOPTWGRASP (TOPTWSolution sol)
- void GRASP (int maxIterations, int maxSizeRCL)

- int aleatorySelectionRCL (int maxTRCL)
- int fuzzySelectionBestFDRCL (ArrayList< double[ ] > rcl)
- int fuzzySelectionAlphaCutRCL (ArrayList< double[ ] > rcl, double alpha)
- void computeGreedySolution (int maxSizeRCL)
- void updateSolution (double[ ] candidateSelected, ArrayList< ArrayList< Double > > departureTimes)
- ArrayList< double[ ] > comprehensiveEvaluation (ArrayList< Integer > customers, ArrayList< ArrayList< Double > > departureTimes)
- TOPTWSolution getSolution ()
- void setSolution (TOPTWSolution solution)
- int getSolutionTime ()
- void setSolutionTime (int solutionTime)
- double getMaxScore ()

Static Public Attributes

- static double NO_EVALUATED = -1.0

Private Attributes

- TOPTWSolution solution
- int solutionTime

6.8.1 Constructor & Destructor Documentation

6.8.1.1 TOPTWGRASP()

```
top.TOPTWGRASP.TOPTWGRASP (
            TOPTWSolution sol )
```

6.8.2 Member Function Documentation

6.8.2.1 aleatorySelectionRCL()

```
int top.TOPTWGRASP.aleatorySelectionRCL (
            int maxTRCL )
```

6.8.2.2 comprehensiveEvaluation()

```
ArrayList< double[] > top.TOPTWGRASP.comprehensiveEvaluation (
            ArrayList< Integer > customers,
            ArrayList< ArrayList< Double > > departureTimes )
```

6.8.2.3 computeGreedySolution()

```
void top.TOPTWGRASP.computeGreedySolution (
            int maxSizeRCL )
```

6.8.2.4 fuzzySelectionAlphaCutRCL()

```
int top.TOPTWGRASP.fuzzySelectionAlphaCutRCL (
            ArrayList< double[] > rcl,
            double alpha )
```

6.8.2.5 fuzzySelectionBestFDRCL()

```
int top.TOPTWGRASP.fuzzySelectionBestFDRCL (
            ArrayList< double[] > rcl )
```

6.8.2.6 getMaxScore()

```
double top.TOPTWGRASP.getMaxScore ( )
```

6.8.2.7 getSolution()

```
TOPTWSolution top.TOPTWGRASP.getSolution ( )
```

6.8.2.8 getSolutionTime()

```
int top.TOPTWGRASP.getSolutionTime ( )
```

6.8.2.9 GRASP()

```
void top.TOPTWGRASP.GRASP (
            int maxIterations,
            int maxSizeRCL )
```

6.8.2.10 setSolution()

```
void top.TOPTWGRASP.setSolution (
            TOPTWSolution solution )
```

6.8.2.11 setSolutionTime()

```
void top.TOPTWGRASP.setSolutionTime (
            int solutionTime )
```

6.8.2.12 updateSolution()

```
void top.TOPTWGRASP.updateSolution (
            double[] candidateSelected,
            ArrayList< ArrayList< Double > > departureTimes )
```

6.8.3 Member Data Documentation

6.8.3.1 NO_EVALUATED

```
double top.TOPTWGRASP.NO_EVALUATED = -1.0  [static]
```

6.8.3.2 solution

```
TOPTWSolution top.TOPTWGRASP.solution  [private]
```

6.8.3.3 solutionTime

```
int top.TOPTWGRASP.solutionTime  [private]
```

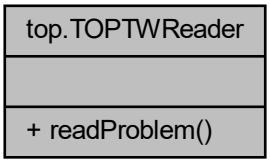
The documentation for this class was generated from the following file:

- src/main/java/top/TOPTWGRASP.java

6.9 top.TOPTWReader Class Reference

Class read a TOPTW problem.

Collaboration diagram for top.TOPTWReader:



Static Public Member Functions

- static TOPTW readProblem (String filePath)
  Read a TOPTW problem from file.

6.9.1 Detailed Description

Class read a TOPTW problem.

6.9.2 Member Function Documentation

6.9.2.1 readProblem()

```
static TOPTW top.TOPTWReader.readProblem (
            String filePath )  [static]
```

Read a TOPTW problem from file.

Parameters

filePath	-> file path

Returns

TOPTW -> TOPTW problem object

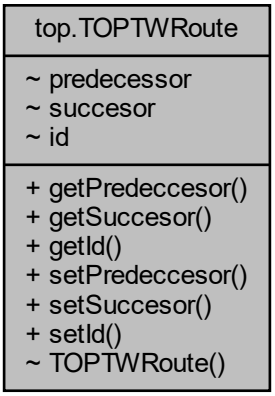
The documentation for this class was generated from the following file:

- src/main/java/top/TOPTWReader.java

6.10 top.TOPTWRoute Class Reference

Class to represent the route.

Collaboration diagram for top.TOPTWRoute:



Public Member Functions

- int getPredeccesor ()
  Getter.
- int getSuccesor ()
  Getter.
- int getId ()
  Getter.
- void setPredeccesor (int pre)
  Setter.
- void setSuccesor (int suc)
  Setter.
- void setId (int id)
  Setter.

6.10.1 Detailed Description

Class to represent the route.

6.10.2 Member Function Documentation

6.10.2.1 getId()

```
int top.TOPTWRoute.getId ( )
```

Getter.

Returns

int -> route id

6.10.2.2 getPredeccesor()

```
int top.TOPTWRoute.getPredeccesor ( )
```

Getter.

Returns

int -> predecessor

6.10.2.3 getSuccesor()

```
int top.TOPTWRoute.getSuccesor ( )
```

Getter.

Returns

int -> sucessor

6.10.2.4 setId()

```
void top.TOPTWRoute.setId (
            int id )
```

Setter.

Parameters

id	-> route id

6.10.2.5 setPredeccesor()

```
void top.TOPTWRoute.setPredeccesor (
            int pre )
```

Setter.

Parameters

pre	-> predecessor

6.10.2.6 setSuccesor()

```
void top.TOPTWRoute.setSuccesor (
            int suc )
```

Setter.

Parameters

suc	-> sucessor

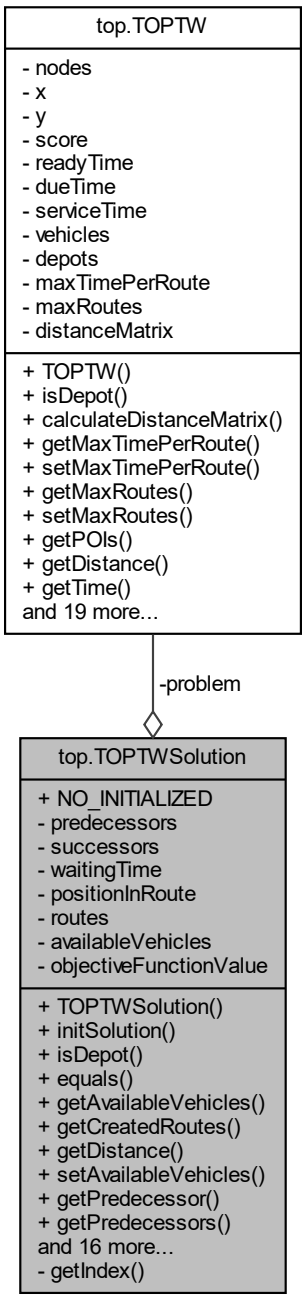
The documentation for this class was generated from the following file:

- src/main/java/top/TOPTWRoute.java

6.11 top.TOPTWSolution Class Reference

Class to represent the TOPTW problem solution.

Collaboration diagram for top.TOPTWSolution:



Public Member Functions

- TOPTWSolution (TOPTW problem)
- void initSolution ()
- boolean isDepot (int c)
- boolean equals (TOPTWSolution otherSolution)
- int getAvailableVehicles ()

- int getCreatedRoutes ()
- double getDistance (int x, int y)
- void setAvailableVehicles (int availableVehicles)
- int getPredecessor (int customer)
- int[ ] getPredecessors ()
- TOPTW getProblem ()
- double getObjectiveFunctionValue ()
- int getPositionInRoute (int customer)
- int getSuccessor (int customer)
- int[ ] getSuccessors ()
- int getIndexRoute (int index)
- double getWaitingTime (int customer)
- void setObjectiveFunctionValue (double objectiveFunctionValue)
- void setPositionInRoute (int customer, int position)
- void setPredecessor (int customer, int predecessor)
- void setSuccessor (int customer, int succesor)
- void setWaitingTime (int customer, int waitingTime)
- String getInfoSolution ()
- double evaluateFitness ()
- int addRoute ()
- double printSolution ()

Static Public Attributes

- static final int NO_INITIALIZED = -1

Private Member Functions

- int getIndex (String[ ] strings, int suc, int index)

Private Attributes

- final TOPTW problem
- int[ ] predecessors
- int[ ] successors
- final double[ ] waitingTime
- final int[ ] positionInRoute
- int[ ] routes
- int availableVehicles
- double objectiveFunctionValue

6.11.1 Detailed Description

Class to represent the TOPTW problem solution.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 TOPTWSolution()

```
top.TOPTWSolution.TOPTWSolution (
            TOPTW problem )
```

6.11.3 Member Function Documentation

6.11.3.1 addRoute()

```
int top.TOPTWSolution.addRoute ( )
```

6.11.3.2 equals()

```
boolean top.TOPTWSolution.equals (
            TOPTWSolution otherSolution )
```

6.11.3.3 evaluateFitness()

```
double top.TOPTWSolution.evaluateFitness ( )
```

6.11.3.4 getAvailableVehicles()

```
int top.TOPTWSolution.getAvailableVehicles ( )
```

6.11.3.5 getCreatedRoutes()

```
int top.TOPTWSolution.getCreatedRoutes ( )
```

6.11.3.6 getDistance()

```
double top.TOPTWSolution.getDistance (
            int x,
            int y )
```

6.11.3.7 getIndex()

```
int top.TOPTWSolution.getIndex (
            String[] strings,
            int suc,
            int index )  [private]
```

6.11.3.8 getIndexRoute()

```
int top.TOPTWSolution.getIndexRoute (
            int index )
```

6.11.3.9 getInfoSolution()

```
String top.TOPTWSolution.getInfoSolution ( )
```

6.11.3.10 getObjectiveFunctionValue()

```
double top.TOPTWSolution.getObjectiveFunctionValue ( )
```

6.11.3.11 getPositionInRoute()

```
int top.TOPTWSolution.getPositionInRoute (
            int customer )
```

6.11.3.12 getPredecessor()

```
int top.TOPTWSolution.getPredecessor (
            int customer )
```

6.11.3.13 getPredecessors()

```
int [] top.TOPTWSolution.getPredecessors ( )
```

6.11.3.14 getProblem()

```
TOPTW top.TOPTWSolution.getProblem ( )
```

6.11.3.15 getSuccessor()

```
int top.TOPTWSolution.getSuccessor (
            int customer )
```

6.11.3.16 getSuccessors()

```
int [] top.TOPTWSolution.getSuccessors ( )
```

6.11.3.17 getWaitingTime()

```
double top.TOPTWSolution.getWaitingTime (
            int customer )
```

6.11.3.18 initSolution()

```
void top.TOPTWSolution.initSolution ( )
```

6.11.3.19 isDepot()

```
boolean top.TOPTWSolution.isDepot (
            int c )
```

6.11.3.20 printSolution()

```
double top.TOPTWSolution.printSolution ( )
```

6.11.3.21 setAvailableVehicles()

```
void top.TOPTWSolution.setAvailableVehicles (
            int availableVehicles )
```

6.11.3.22 setObjectiveFunctionValue()

```
void top.TOPTWSolution.setObjectiveFunctionValue (
            double objectiveFunctionValue )
```

6.11.3.23 setPositionInRoute()

```
void top.TOPTWSolution.setPositionInRoute (
            int customer,
            int position )
```

6.11.3.24 setPredecessor()

```
void top.TOPTWSolution.setPredecessor (
            int customer,
            int predecessor )
```

6.11.3.25 setSuccessor()

```
void top.TOPTWSolution.setSuccessor (
            int customer,
            int succesor )
```

6.11.3.26 setWaitingTime()

```
void top.TOPTWSolution.setWaitingTime (
            int customer,
            int waitingTime )
```

6.11.4 Member Data Documentation

6.11.4.1 availableVehicles

```
int top.TOPTWSolution.availableVehicles  [private]
```

6.11.4.2 NO_INITIALIZED

```
final int top.TOPTWSolution.NO_INITIALIZED = -1  [static]
```

6.11.4.3 objectiveFunctionValue

```
double top.TOPTWSolution.objectiveFunctionValue  [private]
```

6.11.4.4 positionInRoute

```
final int [] top.TOPTWSolution.positionInRoute  [private]
```

6.11.4.5 predecessors

```
int [] top.TOPTWSolution.predecessors  [private]
```

6.11.4.6 problem

```
final TOPTW top.TOPTWSolution.problem  [private]
```

6.11.4.7 routes

```
int [] top.TOPTWSolution.routes  [private]
```

6.11.4.8 successors

```
int [] top.TOPTWSolution.successors  [private]
```

6.11.4.9 waitingTime

```
final double [] top.TOPTWSolution.waitingTime  [private]
```

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

- src/main/java/top/TOPTWSolution.java

Chapter 7

File Documentation

7.1 src/main/java/es/ull/esit/utilities/BellmanFord.java File Reference

Classes

- class es.ull.esit.utilities.BellmanFord

  Implementation of BellmanFord algorithm.

Packages

- package es.ull.esit.utilities

7.2 src/main/java/es/ull/esit/utilities/ExpositoUtilities.java File Reference

Classes

- class es.ull.esit.utilities.ExpositoUtilities

  Different auxiliary utilities to be used along the project.

Packages

- package es.ull.esit.utilities

7.3 src/main/java/es/ull/esit/utilities/PowerSet.java File Reference

Classes

- class es.ull.esit.utilities.PowerSet< E >

  Class to calculate every subset of a given set.

Packages

- package es.ull.esit.utilities

7.4 src/main/java/es/ull/esit/utils/Pair.java File Reference

Classes

- class es.ull.esit.utils.Pair< F, S >

  Class to represent a generic pair of objects.

Packages

- package es.ull.esit.utils

7.5 src/main/java/top/mainTOPTW.java File Reference

Classes

- class top.mainTOPTW

  Main program class.

Packages

- package top

7.6 src/main/java/top/TOPTW.java File Reference

Classes

- class top.TOPTW

Packages

- package top

7.7 src/main/java/top/TOPTWEvaluator.java File Reference

Classes

- class top.TOPTWEvaluator

Packages

- package top

7.8 src/main/java/top/TOPTWGRASP.java File Reference

Classes

- class top.TOPTWGRASP

Packages

- package top

7.9 src/main/java/top/TOPTWReader.java File Reference

Classes

- class top.TOPTWReader
  Class read a TOPTW problem.

Packages

- package top

7.10 src/main/java/top/TOPTWRoute.java File Reference

Classes

- class top.TOPTWRoute
  Class to represent the route.

Packages

- package top

7.11 src/main/java/top/TOPTWSolution.java File Reference

Classes

- class top.TOPTWSolution
  Class to represent the TOPTW problem solution.

Packages

- package top