

ExpositoTOP

Generated by Doxygen 1.9.8

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

es.ull.esit.utilities.BellmanFord	??
es.ull.esit.utilities.ExpositoUtilities	??
Iterable	
es.ull.esit.utilities.PowerSet< E >	??
Iterator	
es.ull.esit.utilities.PowerSet< E >	??
top.mainTOPTW	??
es.ull.esit.utils.Pair< F, S >	??
top.TOPTW	??
top.TOPTWEvaluator	??
top.TOPTWGRASP	??
top.TOPTWReader	??
top.TOPTWRoute	??
top.TOPTWSolution	??

Chapter 2

Class Index

2.1 Class List

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

es.ull.esit.utilities.BellmanFord	Implementa el algoritmo de Bellman-Ford para encontrar el camino más corto en un grafo . . .	??
es.ull.esit.utilities.ExpositoUtilities	Contiene métodos de utilidad para operaciones diversas	??
top.mainTOPTW	Clase que contiene el método main para ejecutar el algoritmo GRASP en diferentes instancias del problema TOPTW	??
es.ull.esit.utils.Pair< F, S >		??
es.ull.esit.utilities.PowerSet< E >	Calcula todos los subconjuntos de un conjunto dado	??
top.TOPTW	Implementación del problema de rutas de vehículos con ventana de tiempo (TOPTW)	??
top.TOPTWEvaluator	Implementación del evaluador de soluciones TOPTW	??
top.TOPTWGRASP	Implementación del algoritmo GRASP para el problema TOPTW	??
top.TOPTWReader	Implementación del lector de instancias TOPTW	??
top.TOPTWRoute	Representa una ruta en el problema TOPTW	??
top.TOPTWSolution	Representa una solución al problema TOPTW	??

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

es/ull/esit/utilities/ BellmanFord.java	Contiene la implementación de la clase BellmanFord	??
es/ull/esit/utilities/ ExpositoUtilities.java	Contiene utilidades para operaciones diversas	??
es/ull/esit/utilities/ PowerSet.java	Contiene la definición de la clase PowerSet para calcular todos los subconjuntos de un conjunto dado	??
top/ mainTOPTW.java	Contiene la clase principal que ejecuta el algoritmo GRASP en diferentes instancias del problema TOPTW	??
top/ TOPTW.java	Contiene la implementación del problema TOPTW y sus funciones asociadas	??
top/ TOPTWEvaluator.java	Contiene la implementación del evaluador de soluciones TOPTW	??
top/ TOPTWGRASP.java	Contiene la implementación del algoritmo GRASP para el problema TOPTW	??
top/ TOPTWReader.java	Contiene la implementación del lector de instancias TOPTW	??
top/ TOPTWRoute.java	Contiene la implementación de la clase TOPTWRoute	??
top/ TOPTWSolution.java	Contiene la implementación de la clase TOPTWSolution	??

Chapter 4

Class Documentation

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

Implementa el algoritmo de Bellman-Ford para encontrar el camino más corto en un grafo.

Public Member Functions

- [BellmanFord](#) (int[][] distanceMatrix, int nodes, ArrayList< Integer > path)
- int[] [getDistances](#) ()
- int [getValue](#) ()
- void **solve** ()

4.1.1 Detailed Description

Implementa el algoritmo de Bellman-Ford para encontrar el camino más corto en un grafo.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 BellmanFord()

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

Parameters

<i>distanceMatrix</i>	
<i>nodes</i>	
<i>path</i>	

4.1.3 Member Function Documentation

4.1.3.1 getDistances()

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

Returns

4.1.3.2 getValue()

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

Returns

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

- es/ull/esit/utilities/[BellmanFord.java](#)

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

Contiene métodos de utilidad para operaciones diversas.

Static Public Member Functions

- static void **printFile** (String file)
- static String **simplifyString** (String string)
- static double[][] **multiplyMatrices** (double a[], double b[])
- static void **writeTextToFile** (String file, String text) throws IOException
- static String **getFormat** (String string)
- static String **getFormat** (double value)
- static String **getFormat** (double value, int zeros)
- static String **getFormat** (String string, int width)
- static String **getFormat** (String string, int width, int alignment)
- 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)
- static boolean **isDouble** (String str)
- static boolean **isAcyclic** (int[][] distanceMatrix)
- static boolean **therelsPath** (int[][] distanceMatrix, int node)

Static Public Attributes

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

4.2.1 Detailed Description

Contiene métodos de utilidad para operaciones diversas.

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

- es/ull/esit/utilities/[ExpositoUtilities.java](#)

4.3 top.mainTOPTW Class Reference

Clase que contiene el método main para ejecutar el algoritmo GRASP en diferentes instancias del problema TOPTW.

Static Public Member Functions

- static void [main](#) (String[] args)
Método principal que ejecuta el algoritmo GRASP en diferentes instancias del problema TOPTW.

4.3.1 Detailed Description

Clase que contiene el método main para ejecutar el algoritmo GRASP en diferentes instancias del problema TOPTW.

4.3.2 Member Function Documentation

4.3.2.1 main()

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

Método principal que ejecuta el algoritmo GRASP en diferentes instancias del problema TOPTW.

Parameters

<i>args</i>	Argumentos de la línea de comandos (no se utilizan en este programa).
-------------	---

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

- top/[mainTOPTW.java](#)

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

Public Member Functions

- **Pair** (F first, S second)
- boolean **equals** (Object o)
- int **hashCode** ()

Static Public Member Functions

- static< A, B > **Pair**< A, B > **create** (A a, B b)

Public Attributes

- final F **first**
- final S **second**

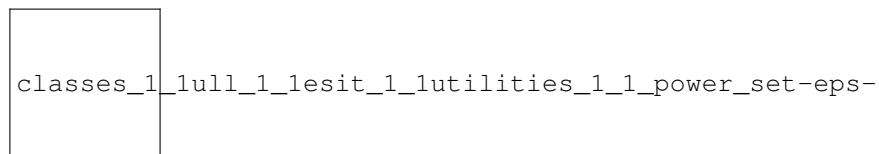
The documentation for this class was generated from the following file:

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

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

Calcula todos los subconjuntos de un conjunto dado.

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



Public Member Functions

- **PowerSet** (Set< E > set)
Constructor de la clase PowerSet.
- boolean **hasNext** ()
Comprueba si hay más subconjuntos por calcular.
- Set< E > **next** ()
Obtiene el siguiente subconjunto.
- void **remove** ()
No se permite la operación de eliminación.
- Iterator< Set< E > > **iterator** ()
Obtiene un iterador para los subconjuntos.

4.5.1 Detailed Description

Calcula todos los subconjuntos de un conjunto dado.

Template Parameters

<i>E</i>	Tipo de elementos del conjunto.
----------	---------------------------------

4.5.2 Constructor & Destructor Documentation

4.5.2.1 PowerSet()

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

Constructor de la clase PowerSet.

Parameters

<i>set</i>	Conjunto del cual se calcularán los subconjuntos.
------------	---

4.5.3 Member Function Documentation

4.5.3.1 hasNext()

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

Comprueba si hay más subconjuntos por calcular.

Returns

true si hay más subconjuntos, false en caso contrario.

4.5.3.2 iterator()

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

Obtiene un iterador para los subconjuntos.

Returns

Iterador de subconjuntos.

4.5.3.3 next()

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

Obtiene el siguiente subconjunto.

Returns

Conjunto que representa el subconjunto actual.

4.5.3.4 remove()

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

No se permite la operación de eliminación.

Exceptions

<i>UnsupportedOperationException</i>	Si se llama a este método.
--------------------------------------	----------------------------

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

- es/ull/esit/utilities/[PowerSet.java](#)

4.6 top.TOPTW Class Reference

Implementación del problema de rutas de vehículos con ventana de tiempo (TOPTW).

Public Member Functions

- [TOPTW](#) (int nodes, int routes)
Constructor de la clase TOPTW.
- boolean **isDepot** (int a)
- double **getDistance** (int[] route)
- double **getDistance** (ArrayList< Integer > route)
- double **getDistance** (ArrayList< Integer >[] routes)
- 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** ()

4.6.1 Detailed Description

Implementación del problema de rutas de vehículos con ventana de tiempo (TOPTW).

4.6.2 Constructor & Destructor Documentation

4.6.2.1 TOPTW()

```
top.TOPTW.TOPTW (
    int nodes,
    int routes ) [inline]
```

Constructor de la clase TOPTW.

Parameters

<i>nodes</i>	Número de nodos.
<i>routes</i>	Número de rutas.

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

- [top/TOPTW.java](#)

4.7 top.TOPTWEvaluator Class Reference

Implementación del evaluador de soluciones TOPTW.

Public Member Functions

- void [evaluate](#) ([TOPTWSolution](#) solution)
Evalúa una solución TOPTW.

Static Public Attributes

- static double **NO_EVALUATED** = -1.0

4.7.1 Detailed Description

Implementación del evaluador de soluciones TOPTW.

4.7.2 Member Function Documentation

4.7.2.1 evaluate()

```
void top.TOPTWEvaluator.evaluate (
    TOPTWSolution solution ) [inline]
```

Evalúa una solución TOPTW.

Parameters

<code>solution</code>	La solución a evaluar.
-----------------------	------------------------

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

- [top/TOPTWEvaluator.java](#)

4.8 top.TOPTWGRASP Class Reference

Implementación del algoritmo GRASP para el problema TOPTW.

Public Member Functions

- [TOPTWGRASP](#) ([TOPTWSolution](#) sol)
Constructor de la clase TOPTWGRASP.
- 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

4.8.1 Detailed Description

Implementación del algoritmo GRASP para el problema TOPTW.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 TOPTWGRASP()

```
top.TOPTWGRASP.TOPTWGRASP (
    TOPTWSolution sol ) [inline]
```

Constructor de la clase TOPTWGRASP.

Parameters

<code>sol</code>	Una solución inicial para el problema TOPTW.
------------------	--

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

- [top/TOPTWGRASP.java](#)

4.9 top.TOPTWReader Class Reference

Implementación del lector de instancias TOPTW.

Static Public Member Functions

- static [TOPTW readProblem](#) (String filePath)
Lee una instancia del problema TOPTW desde un archivo.

4.9.1 Detailed Description

Implementación del lector de instancias TOPTW.

4.9.2 Member Function Documentation

4.9.2.1 readProblem()

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

Lee una instancia del problema TOPTW desde un archivo.

Parameters

<code>filePath</code>	Ruta del archivo de la instancia.
-----------------------	-----------------------------------

Returns

Objeto TOPTW con la instancia cargada.

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

- [top/TOPTWReader.java](#)

4.10 top.TOPTWRoute Class Reference

Representa una ruta en el problema TOPTW.

Public Member Functions

- int [getPredecessor](#) ()
Obtiene el predecesor de la ruta.
- int [getSuccessor](#) ()
Obtiene el sucesor de la ruta.
- int [getId](#) ()
Obtiene el identificador de la ruta.
- void [setPredecessor](#) (int pre)
Establece el predecesor de la ruta.
- void [setSuccessor](#) (int suc)
Establece el sucesor de la ruta.
- void [setId](#) (int id)
Establece el identificador de la ruta.

4.10.1 Detailed Description

Representa una ruta en el problema TOPTW.

4.10.2 Member Function Documentation

4.10.2.1 [getId\(\)](#)

```
int top.TOPTWRoute.getId ( ) [inline]
```

Obtiene el identificador de la ruta.

Returns

El identificador.

4.10.2.2 [getPredecessor\(\)](#)

```
int top.TOPTWRoute.getPredecessor ( ) [inline]
```

Obtiene el predecesor de la ruta.

Returns

El predecesor.

4.10.2.3 [getSuccessor\(\)](#)

```
int top.TOPTWRoute.getSuccessor ( ) [inline]
```

Obtiene el sucesor de la ruta.

Returns

El sucesor.

4.10.2.4 [setId\(\)](#)

```
void top.TOPTWRoute.setId (  
    int id ) [inline]
```

Establece el identificador de la ruta.

Parameters

<i>id</i>	El nuevo identificador.
-----------	-------------------------

4.10.2.5 setPredecessor()

```
void top.TOPTWRoute.setPredecessor (
    int pre ) [inline]
```

Establece el predecesor de la ruta.

Parameters

<i>pre</i>	El nuevo predecesor.
------------	----------------------

4.10.2.6 setSuccessor()

```
void top.TOPTWRoute.setSuccessor (
    int suc ) [inline]
```

Establece el sucesor de la ruta.

Parameters

<i>suc</i>	El nuevo sucesor.
------------	-------------------

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

- [top/TOPTWRoute.java](#)

4.11 top.TOPTWSolution Class Reference

Representa una solución al problema TOPTW.

Public Member Functions

- [TOPTWSolution](#) ([TOPTW](#) problem)
Constructor de TOPTWSolution.
- 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 sucesor)
- void **setWaitingTime** (int customer, int waitingTime)
- String **getInfoSolution** ()
- double **evaluateFitness** ()
- int **addRoute** ()
- double **printSolution** ()

Static Public Attributes

- static final int **NO_INITIALIZED** = -1

4.11.1 Detailed Description

Representa una solución al problema TOPTW.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 TOPTWSolution()

```
top.TOPTWSolution.TOPTWSolution (
    TOPTW problem ) [inline]
```

Constructor de TOPTWSolution.

Parameters

<i>problem</i>	El problema asociado a la solución.
----------------	-------------------------------------

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

- [top/TOPTWSolution.java](#)

Chapter 5

File Documentation

5.1 [es/ull/esit/utilities/BellmanFord.java](#) File Reference

Contiene la implementación de la clase BellmanFord.

```
import java.util.ArrayList;
```

Classes

- class [es.ull.esit.utilities.BellmanFord](#)
Implementa el algoritmo de Bellman-Ford para encontrar el camino más corto en un grafo.

5.1.1 Detailed Description

Contiene la implementación de la clase BellmanFord.

5.2 [es/ull/esit/utilities/ExpositoUtilities.java](#) File Reference

Contiene utilidades para operaciones diversas.

```
import java.io.BufferedReader;
```

Classes

- class [es.ull.esit.utilities.ExpositoUtilities](#)
Contiene métodos de utilidad para operaciones diversas.

5.2.1 Detailed Description

Contiene utilidades para operaciones diversas.

5.3 es/ull/esit/utilities/PowerSet.java File Reference

Contiene la definición de la clase PowerSet para calcular todos los subconjuntos de un conjunto dado.

```
import java.util.BitSet;
```

Classes

- class [es.ull.esit.utilities.PowerSet< E >](#)
Calcula todos los subconjuntos de un conjunto dado.

5.3.1 Detailed Description

Contiene la definición de la clase PowerSet para calcular todos los subconjuntos de un conjunto dado.

5.4 top/mainTOPTW.java File Reference

Contiene la clase principal que ejecuta el algoritmo GRASP en diferentes instancias del problema TOPTW.

Classes

- class [top.mainTOPTW](#)
Clase que contiene el método main para ejecutar el algoritmo GRASP en diferentes instancias del problema TOPTW.

5.4.1 Detailed Description

Contiene la clase principal que ejecuta el algoritmo GRASP en diferentes instancias del problema TOPTW.

5.5 top/TOPTW.java File Reference

Contiene la implementación del problema TOPTW y sus funciones asociadas.

```
import java.util.ArrayList;
```

Classes

- class [top.TOPTW](#)
Implementación del problema de rutas de vehículos con ventana de tiempo (TOPTW).

5.5.1 Detailed Description

Contiene la implementación del problema TOPTW y sus funciones asociadas.

5.6 top/TOPTWEvaluator.java File Reference

Contiene la implementación del evaluador de soluciones TOPTW.

Classes

- class [top.TOPTWEvaluator](#)
Implementación del evaluador de soluciones TOPTW.

5.6.1 Detailed Description

Contiene la implementación del evaluador de soluciones TOPTW.

5.7 top/TOPTWGRASP.java File Reference

Contiene la implementación del algoritmo GRASP para el problema TOPTW.

```
import java.util.ArrayList;
```

Classes

- class [top.TOPTWGRASP](#)
Implementación del algoritmo GRASP para el problema TOPTW.

5.7.1 Detailed Description

Contiene la implementación del algoritmo GRASP para el problema TOPTW.

5.8 top/TOPTWReader.java File Reference

Contiene la implementación del lector de instancias TOPTW.

```
import java.io.BufferedReader;
```

Classes

- class [top.TOPTWReader](#)
Implementación del lector de instancias TOPTW.

5.8.1 Detailed Description

Contiene la implementación del lector de instancias TOPTW.

5.9 top/TOPTWRoute.java File Reference

Contiene la implementación de la clase TOPTWRoute.

Classes

- class [top.TOPTWRoute](#)
Representa una ruta en el problema TOPTW.

5.9.1 Detailed Description

Contiene la implementación de la clase TOPTWRoute.

5.10 top/TOPTWSolution.java File Reference

Contiene la implementación de la clase TOPTWSolution.

```
import java.util.Arrays;
```

Classes

- class [top.TOPTWSolution](#)
Representa una solución al problema TOPTW.

5.10.1 Detailed Description

Contiene la implementación de la clase TOPTWSolution.