My Project

Generated by Doxygen 1.12.0

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

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

Class to solve the Bellman-Ford algorithm.

Public Member Functions

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

  Constructor of the class.
- int[ ] getDistances ()

  Method to get the distances between nodes.
- int getValue ()

  Method to get the value of the path between the nodes.
- void **solve** ()

  Method to solve the problem.

3.1.1 Detailed Description

Class to solve the Bellman-Ford algorithm.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 BellmanFord()

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

Constructor of the class.

Parameters

distanceMatrix	
nodes	
path	

3.1.3 Member Function Documentation

3.1.3.1 getDistances()

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

Method to get the distances between nodes.

Returns

Array with the distances between nodes.

3.1.3.2 getValue()

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

Method to get the value of the path between the nodes.

Returns

Value of the path between the nodes.

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

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

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

Class to store the utilities of the project.

Static Public Member Functions

- static void printFile (String file)

    Method to get the last appearance of an element in a vector.
- static String simplifyString (String string)

    Method to simplify a string.
- static double[ ][ ] multiplyMatrices (double a[ ][ ], double b[ ][ ])

    Method to multiply two matrices.
- static void writeTextToFile (String file, String text) throws IOException

    Method to write a text to a file.
- static String getFormat (String string)

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

    Method to get the format of a double.
- static String getFormat (double value, int zeros)

    Method to get the format of a double.
- static String getFormat (String string, int width)

    Method to get the format of a string.
- static String getFormat (String string, int width, int alignment)

    Method to get the format of a string.
- static String getFormat (ArrayList< String > strings, int width)

    Method to get the format of a string.
- static String getFormat (ArrayList< Integer > strings)

    Method to get the format of a string.
- static String getFormat (String[ ] strings, int width)

    Method to get the format of a string.
- static String getFormat (String[ ][ ] matrixStrings, int width)

    Method to get the format of a string.
- static String getFormat (String[ ] strings, int[ ] width)

    Method to get the format of a string.
- static String getFormat (String[ ] strings)

    Method to get the format of a string.
- static String getFormat (String[ ] strings, int[ ] width)

    Method to get the format of a string.
- static String getFormat (String[ ] strings, int[ ] width, int[ ] alignment)

    Method to get the format of a string.
- static boolean isInteger (String str)

    Method to know if a string is an integer.
- static boolean isDouble (String str)

    Method to know if a string is a double.
- static boolean isAcyclic (int[ ][ ] distanceMatrix)

    Method to know is a matrix is acyclic.
- static boolean thereIsPath (int[ ][ ] distanceMatrix, int node)

    Method to know if there is a path between two nodes.

Static Public Attributes

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

3.2.1 Detailed Description

Class to store the utilities of the project.

3.2.2 Member Function Documentation

3.2.2.1 getFormat() [1/13]

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

Method to get the format of a string.

Parameters

strings	

Returns

The format of the string.

3.2.2.2 getFormat() [2/13]

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

Method to get the format of a string.

Parameters

strings	
width	

Returns

The format of the string.

3.2.2.3 getFormat() [3/13]

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

Method to get the format of a double.

Parameters

value	

Returns

The format of the double.

3.2.2.4 getFormat() [4/13]

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

Method to get the format of a double.

Parameters

value	
zeros	

Returns

The format of the double.

3.2.2.5 getFormat() [5/13]

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

Method to get the format of a string.

Parameters

string	

Returns

The format of the string.

3.2.2.6 getFormat() [6/13]

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

Method to get the format of a string.

Parameters

string	
width	

Returns

The format of the string.

3.2.2.7 getFormat() [7/13]

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

Method to get the format of a string.

Parameters

string	
width	
alignment	

Returns

The format of the string.

3.2.2.8 getFormat() [8/13]

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

Method to get the format of a string.

Parameters

strings	

Returns

The format of the string.

3.2.2.9 getFormat() [9/13]

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

Method to get the format of a string.

Parameters

strings	
width	

Returns

The format of the string.

3.2.2.10 getFormat() [10/13]

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

Method to get the format of a string.

Parameters

strings	

Returns

The format of the string.

3.2.2.11 getFormat() [11/13]

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

Method to get the format of a string.

Parameters

strings	
width	
alignment	

Returns

The format of the string.

3.2.2.12 getFormat() [12/13]

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

Method to get the format of a string.

Parameters

strings	
width	
alignment	

Returns

The format of the string.

3.2.2.13 getFormat() [13/13]

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

Method to get the format of a string.

Parameters

strings	
width	

Returns

The format of the string.

3.2.2.14 isAcyclic()

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

Method to know is a matrix is acyclic.

Parameters

distanceMatrix	

Returns

True if the matrix is acyclic, false otherwise.

3.2.2.15 isDouble()

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

Method to know if a string is a double.

Parameters

str	

Returns

True if the string is a double, false otherwise.

3.2.2.16 isInteger()

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

Method to know if a string is an integer.

Parameters

str	

Returns

True if the string is an integer, false otherwise.

3.2.2.17 multiplyMatrices()

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

Method to multiply two matrices.

Parameters

a	
b	

Returns

The result of the multiplication of the matrices.

3.2.2.18 printFile()

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

Method to get the last appearance of an element in a vector.

Parameters

vector	
element	

Returns

The last appearance of the element in the vector.

3.2.2.19 simplifyString()

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

Method to simplify a string.

Parameters

string	

Returns

The simplified string.

3.2.2.20 thereIsPath()

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

Method to know if there is a path between two nodes.

Parameters

distanceMatrix	
node	

Returns

True if there is a path between the nodes, false otherwise.

3.2.2.21 writeTextToFile()

```
static void es.ull.esit.utilities.ExpositoUtilities.writeTextToFile (
             String file,
             String text) throws IOException  [static]
```

Method to write a text to a file.

Parameters

file	
text	

Exceptions

IOException	

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

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

3.3 top.mainTOPTW Class Reference

Main class to execute the program.

Static Public Member Functions

- static void main (String[ ] args)

  Main method to execute the program.

3.3.1 Detailed Description

Main class to execute the program.

3.3.2 Member Function Documentation

3.3.2.1 main()

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

Main method to execute the program.

Parameters

args	

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

- src/main/top/mainTOPTW.java

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

Class to create a pair of objects.

Public Member Functions

- Pair (F first, S second)

  Constructor of the class.
- boolean equals (Object o)

  Method to know if two objects are equal.
- int hashCode ()

  Method to calculate the hash code of the object.

Static Public Member Functions

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

  Method to create a pair.

Public Attributes

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

3.4.1 Detailed Description

Class to create a pair of objects.

Parameters

<F>	
<S>	

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Pair()

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

Constructor of the class.

Parameters

first	
second	

3.4.3 Member Function Documentation

3.4.3.1 create()

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

Method to create a pair.

Parameters

a	
b	

Returns

Pair object.

3.4.3.2 equals()

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

Method to know if two objects are equal.

Returns

boolean value.

3.4.3.3 hashCode()

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

Method to calculate the hash code of the object.

Returns

int value.

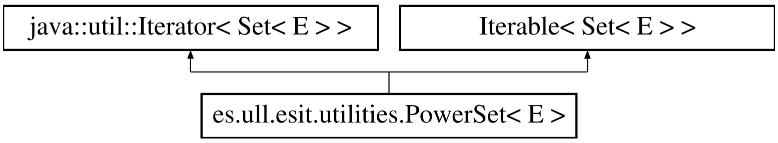
The documentation for this class was generated from the following file:

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

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

Class to calculate the power set of a given set.

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


java::util::Iterator< Set< E > >
Iterable< Set< E > >
es.ull.esit.utilities.PowerSet< E >

Public Member Functions

- PowerSet (Set< E > set)

  Constructor of the class.
- boolean **hasNext** ()

  Method to calculate the edges of the graph.
- Set< E > **next** ()

  Method to calculate the edges of the graph.
- void **remove** ()

  Method to calculate the edges of the graph.
- Iterator< Set< E > > **iterator** ()

  Method to calculate the edges of the graph.

3.5.1 Detailed Description

Class to calculate the power set of a given set.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 PowerSet()

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

Constructor of the class.

Parameters

set	

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

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

3.6 top.TOPTW Class Reference

Class to represent the TOPTW problem.

Public Member Functions

- TOPTW (int nodes, int routes)

  Constructor of the class.
- boolean isDepot (int a)

  Method to know if a node is a depot.
- double getDistance (int[ ] route)

  Method to get the distance of a route.
- double getDistance (ArrayList< Integer > route)

  Method to get the distance of a route.
- double getDistance (ArrayList< Integer >[ ] routes)

  Method to get the distance of a route.
- void **calculateDistanceMatrix** ()

  Method to calculate the distance matrix.
- double **getMaxTimePerRoute** ()

  Method to get the maximum time per route.
- void setMaxTimePerRoute (double maxTimePerRoute)

  Method to set the maximum time per route.
- double **getMaxRoutes** ()

  Method to get the maximum number of routes.
- void setMaxRoutes (double maxRoutes)

  Method to set the maximum number of routes.
- int getPOIs ()

  Method to get the POIs.
- double getDistance (int i, int j)

  Method to get the distance between two nodes.
- double getTime (int i, int j)

  Method to get the time between two nodes.
- int getNodes ()

  Method to get the nodes.
- void setNodes (int nodes)

  Method to set the nodes.
- double getX (int index)

  Method to get the x coordinate of a node.
- void setX (int index, double x)

  Method to set the x coordinate of a node.
- double getY (int index)

  Method to get the y coordinate of a node.
- void setY (int index, double y)

  Method to set the y coordinate of a node.
- double getScore (int index)

  Method to get the score of a node.
- double[ ] **getScore** ()

  Method to get the score of a node.
- void setScore (int index, double score)

  Method to set the score of a node.
- double getReadyTime (int index)

  Method to get the ready time of a node.
- void setReadyTime (int index, double readyTime)

  Method to set the ready time of a node.
- double getDueTime (int index)

Method to get the due time of a node.
- void setDueTime (int index, double dueTime)

  Method to set the due time of a node.
- double getServiceTime (int index)

  Method to get the service time of a node.
- void setServiceTime (int index, double serviceTime)

  Method to set the service time of a node.
- int getVehicles ()

  Method to get the vehicles.
- String **toString** ()

  Method to convert the TOPTW object to a string.
- int addNode ()

  Method add a node.
- int addNodeDepot ()

  Method to add a node to the depot.

3.6.1 Detailed Description

Class to represent the TOPTW problem.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 TOPTW()

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

Constructor of the class.

Parameters

nodes	
routes	

3.6.3 Member Function Documentation

3.6.3.1 addNode()

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

Method add a node.

Returns

int value.

3.6.3.2 addNodeDepot()

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

Method to add a node to the depot.

Returns

int value.

3.6.3.3 getDistance() [1/4]

```
double top.TOPTW.getDistance (
            ArrayList< Integer > route)
```

Method to get the distance of a route.

Parameters

route	

Returns

3.6.3.4 getDistance() [2/4]

```
double top.TOPTW.getDistance (
            ArrayList< Integer >[] routes)
```

Method to get the distance of a route.

Parameters

routes	

Returns

3.6.3.5 getDistance() [3/4]

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

Method to get the distance between two nodes.

Parameters

i	
j	

Returns

double value.

3.6.3.6 getDistance() [4/4]

```
double top.TOPTW.getDistance (
             int[] route)
```

Method to get the distance of a route.

Parameters

route	

Returns

3.6.3.7 getDueTime()

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

Method to get the due time of a node.

Parameters

index	

Returns

double value.

3.6.3.8 getNodes()

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

Method to get the nodes.

Returns

int value.

3.6.3.9 getPOIs()

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

Method to get the POIs.

Returns

3.6.3.10 getReadyTime()

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

Method to get the ready time of a node.

Parameters

index	

Returns

double value.

3.6.3.11 getScore()

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

Method to get the score of a node.

Parameters

index	

Returns

double value.

3.6.3.12 getServiceTime()

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

Method to get the service time of a node.

Parameters

index	

Returns

double value.

3.6.3.13 getTime()

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

Method to get the time between two nodes.

Parameters

i	
j	

Returns

double value.

3.6.3.14 getVehicles()

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

Method to get the vehicles.

Returns

int value.

3.6.3.15 getX()

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

Method to get the x coordinate of a node.

Parameters

index	

Returns

double value.

3.6.3.16 getY()

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

Method to get the y coordinate of a node.

Parameters

index	

Returns

double value.

3.6.3.17 isDepot()

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

Method to know if a node is a depot.

Parameters

a	

Returns

3.6.3.18 setDueTime()

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

Method to set the due time of a node.

Parameters

index	
dueTime	

3.6.3.19 setMaxRoutes()

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

Method to set the maximum number of routes.

Parameters

maxRoutes	

3.6.3.20 setMaxTimePerRoute()

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

Method to set the maximum time per route.

Parameters

maxTimePerRoute	

3.6.3.21 setNodes()

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

Method to set the nodes.

Parameters

nodes	

3.6.3.22 setReadyTime()

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

Method to set the ready time of a node.

Parameters

index	
readyTime	

3.6.3.23 setScore()

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

Method to set the score of a node.

Parameters

index	
score	

3.6.3.24 setServiceTime()

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

Method to set the service time of a node.

Parameters

index	
serviceTime	

3.6.3.25 setX()

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

Method to set the x coordinate of a node.

Parameters

index	
x	

3.6.3.26 setY()

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

Method to set the y coordinate of a node.

Parameters

index	
y	

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

- src/main/top/TOPTW.java

3.7 top.TOPTWEvaluator Class Reference

Class to evaluate the solution of the TOPTW problem.

Public Member Functions

- void evaluate (TOPTWSolution solution)

  Method to evaluate the solution of the TOPTW problem.

Static Public Attributes

- static double **NO_EVALUATED** = -1.0

3.7.1 Detailed Description

Class to evaluate the solution of the TOPTW problem.

3.7.2 Member Function Documentation

3.7.2.1 evaluate()

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

Method to evaluate the solution of the TOPTW problem.

Parameters

solution	

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

- src/main/top/TOPTWEvaluator.java

3.8 top.TOPTWGRASP Class Reference

Public Member Functions

- TOPTWGRASP (TOPTWSolution sol)

  Constructor of the class.
- void GRASP (int maxIterations, int maxSizeRCL)

  Method to execute the GRASP algorithm.
- int aleatorySelectionRCL (int maxTRCL)

  Method to select a random element from the RCL list.
- int fuzzySelectionBestFDRCL (ArrayList< double[ ] > rcl)

  Method to select the best element from the RCL list.
- int fuzzySelectionAlphaCutRCL (ArrayList< double[ ] > rcl, double alpha)

  Method to select a random element from the RCL list.
- void computeGreedySolution (int maxSizeRCL)

  Method to compute the greedy solution.
- void updateSolution (double[ ] candidateSelected, ArrayList< ArrayList< Double > > departureTimes)

  Method to update the solution.
- ArrayList< double[ ] > comprehensiveEvaluation (ArrayList< Integer > customers, ArrayList< ArrayList< Double > > departureTimes)

  Method to evaluate the comprehensive evaluation of the solution.
- TOPTWSolution getSolution ()

Method to get the solution.
- void setSolution (TOPTWSolution solution)

  Method to set the solution.
- int getSolutionTime ()

  Method to get the solution time.
- void setSolutionTime (int solutionTime)

  Method to set the solution time.
- double getMaxScore ()

  Method to get the max score.

Static Public Attributes

- static double **NO_EVALUATED** = -1.0

3.8.1 Constructor & Destructor Documentation

3.8.1.1 TOPTWGRASP()

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

Constructor of the class.

Parameters

solution	

3.8.2 Member Function Documentation

3.8.2.1 aleatorySelectionRCL()

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

Method to select a random element from the RCL list.

Parameters

maxTRCL	

Returns

3.8.2.2 comprehensiveEvaluation()

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

Method to evaluate the comprehensive evaluation of the solution.

Parameters

customers	
departureTimes	

Returns

3.8.2.3 computeGreedySolution()

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

Method to compute the greedy solution.

Parameters

maxSizeRCL	

3.8.2.4 fuzzySelectionAlphaCutRCL()

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

Method to select a random element from the RCL list.

Parameters

rcl	
alpha	

Returns

3.8.2.5 fuzzySelectionBestFDRCL()

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

Method to select the best element from the RCL list.

Parameters

rcl	

Returns

3.8.2.6 getMaxScore()

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

Method to get the max score.

Returns

3.8.2.7 getSolution()

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

Method to get the solution.

Returns

3.8.2.8 getSolutionTime()

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

Method to get the solution time.

Returns

3.8.2.9 GRASP()

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

Method to execute the GRASP algorithm.

Parameters

maxIterations	
maxSizeRCL	

3.8.2.10 setSolution()

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

Method to set the solution.

Parameters

solution	

3.8.2.11 setSolutionTime()

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

Method to set the solution time.

Parameters

solutionTime	

3.8.2.12 updateSolution()

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

Method to update the solution.

Parameters

candidateSelected	
departureTimes	

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

- src/main/top/TOPTWGRASP.java

3.9 top.TOPTWReader Class Reference

Class to read a TOPTW problem from a file.

Static Public Member Functions

- static TOPTW readProblem (String filePath)

  Read a TOPTW problem from a file.

3.9.1 Detailed Description

Class to read a TOPTW problem from a file.

3.9.2 Member Function Documentation

3.9.2.1 readProblem()

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

Read a TOPTW problem from a file.

Parameters

filePath	Path to the file.

Returns

TOPTW problem.

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

- src/main/top/TOPTWReader.java

3.10 top.TOPTWRoute Class Reference

Class that represents a route in the TOPTW problem.

Public Member Functions

- int getPredeccesor ()

  Get the predecessor of the route.
- int getSuccesor ()

  Get the successor of the route.
- int getId ()

  Get the id of the route.
- void setPredeccesor (int pre)

  Set the predecessor of the route.
- void setSuccesor (int suc)

  Set the successor of the route.
- void setId (int id)

  Set the id of the route.

3.10.1 Detailed Description

Class that represents a route in the TOPTW problem.

3.10.2 Member Function Documentation

3.10.2.1 getId()

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

Get the id of the route.

Returns

Id.

3.10.2.2 getPredeccesor()

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

Get the predecessor of the route.

Returns

Predecessor.

3.10.2.3 getSuccesor()

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

Get the successor of the route.

Returns

Successor.

3.10.2.4 setId()

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

Set the id of the route.

Parameters

id	Id.

3.10.2.5 setPredeccesor()

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

Set the predecessor of the route.

Parameters

pre	Predecessor.

3.10.2.6 setSuccesor()

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

Set the successor of the route.

Parameters

suc	Successor.

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

- src/main/top/TOPTWRoute.java

3.11 top.TOPTWSolution Class Reference

Class to represent a solution for the TOPTW problem.

Public Member Functions

- TOPTWSolution (TOPTW problem)

  Constructor of the class.
- void initSolution ()

  Method to initialize the solution.
- boolean isDepot (int c)

  Method to know if a customer is a depot.
- boolean equals (TOPTWSolution otherSolution)

  Method to know if a customer is a POI.
- int getAvailableVehicles ()

  Method to get the available vehicles.
- int getCreatedRoutes ()

  Method to get the created routes.
- double getDistance (int x, int y)

  Method to get the distance between two nodes.
- void **setAvailableVehicles** (int availableVehicles)

  Method to set the available vehicles.
- int getPredecessor (int customer)

  Method to get predecessor.
- int[ ] getPredecessors ()

  Method to get the predecessors.
- TOPTW getProblem ()

  Method to get the problem.
- double getObjectiveFunctionValue ()

  Method to get the objective function value.
- int getPositionInRoute (int customer)

  Method to get the position in route.
- int getSuccessor (int customer)

  Method to get the successors.
- int[ ] getSuccessors ()

  Method to get the successors.
- int getIndexRoute (int index)

  Method to get the routes. @ param index.
- double getWaitingTime (int customer)

  Method to get the waiting time.

- void setObjectiveFunctionValue (double objectiveFunctionValue)

  Method to set the predecessors.
- void setPositionInRoute (int customer, int position)

  Method to set the position in route.
- void setPredecessor (int customer, int predecessor)

  Method to set the predecessors.
- void setSuccessor (int customer, int succesor)

  Method to set the successors.
- void setWaitingTime (int customer, int waitingTime)

  Method to set the waiting time.
- String getInfoSolution ()

  Method to get the info of the solution.
- double evaluateFitness ()

  Method to evaluate the fitness of the solution.
- int addRoute ()

  Method to add a route to the solution.
- double printSolution ()

  Method to print the solution.

Static Public Attributes

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

3.11.1 Detailed Description

Class to represent a solution for the TOPTW problem.

3.11.2 Constructor & Destructor Documentation

3.11.2.1 TOPTWSolution()

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

Constructor of the class.

Parameters

problem	TOPTW problem to solve.

3.11.3 Member Function Documentation

3.11.3.1 addRoute()

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

Method to add a route to the solution.

Returns

3.11.3.2 equals()

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

Method to know if a customer is a POI.

Parameters

otherSolution	

Returns

True if the customer is a POI, false otherwise.

3.11.3.3 evaluateFitness()

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

Method to evaluate the fitness of the solution.

Returns

3.11.3.4 getAvailableVehicles()

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

Method to get the available vehicles.

Returns

The available vehicles.

3.11.3.5 getCreatedRoutes()

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

Method to get the created routes.

Returns

The created routes.

3.11.3.6 getDistance()

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

Method to get the distance between two nodes.

Parameters

x	
y	

Returns

The distance between the two nodes.

3.11.3.7 getIndexRoute()

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

Method to get the routes. @ param index.

Returns

The routes.

3.11.3.8 getInfoSolution()

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

Method to get the info of the solution.

Returns

The info of the solution.

3.11.3.9 getObjectiveFunctionValue()

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

Method to get the objective function value.

Returns

The objective function value.

3.11.3.10 getPositionInRoute()

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

Method to get the position in route.

Parameters

customer	

Returns

The position in route.

3.11.3.11 getPredecessor()

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

Method to get predecessor.

Parameters

customer	

Returns

3.11.3.12 getPredecessors()

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

Method to get the predecessors.

Returns

The predecessors.

3.11.3.13 getProblem()

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

Method to get the problem.

Returns

The problem.

3.11.3.14 getSuccessor()

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

Method to get the successors.

Parameters

customer	

Returns

The successors.

3.11.3.15 getSuccessors()

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

Method to get the successors.

Returns

The successors.

3.11.3.16 getWaitingTime()

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

Method to get the waiting time.

Parameters

customer	

Returns

The waiting time.

3.11.3.17 initSolution()

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

Method to initialize the solution.

The solution is initialized with the depot as the first node of the first route.

The predecessors and successors arrays are initialized with -1.

The routes array is initialized with -1.

The available vehicles are initialized with the total number of vehicles.

The objective function value is initialized with -1.

The waiting time array is initialized with -1.

The position in route array is initialized with -1.

3.11.3.18 isDepot()

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

Method to know if a customer is a depot.

Parameters

c	

Returns

True if the customer is a depot, false otherwise.

3.11.3.19 printSolution()

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

Method to print the solution.

Returns

The fitness of the solution.

3.11.3.20 setObjectiveFunctionValue()

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

Method to set the predecessors.

Parameters

predecessors	

3.11.3.21 setPositionInRoute()

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

Method to set the position in route.

Parameters

customer	
position	

3.11.3.22 setPredecessor()

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

Method to set the predecessors.

Parameters

customer	
predecessor	

3.11.3.23 setSuccessor()

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

Method to set the successors.

Parameters

customer	
successor	

3.11.3.24 setWaitingTime()

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

Method to set the waiting time.

Parameters

customer	
waitingTime	

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

- src/main/top/TOPTWSolution.java

Index