Route Finder

Generated by Doxygen 1.8.20

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

Class Documentation

2.1 RouteFinder Class Reference

Public Member Functions

- RouteFinder ()
- RouteFinder (const RouteFinder &r)
- RouteFinder (const std::string &f)
- RouteFinder & operator= (RouteFinder x)
- int getHouseNum () const
- std::vector< dLinkedList< double > > getDist () const
- std::vector< int > getBoxes () const
- int getMaxBoxes () const
- double getMaxDist () const
- void setHouseNum (int n)
- void setDist (std::vector< dLinkedList< double > > d)
- void setBoxes (std::vector< int > b)
- void setMaxBoxes (int m)
- void setMaxDist (double d)
- void solve ()
- std::vector< dLinkedList< int > > getRoutes () const
- std::vector< double > getRouteDist () const
- std::vector< int > getBoxNum () const
- double getFinalDist () const

2.1.1 Constructor & Destructor Documentation

2.1.1.1 RouteFinder() `[1/3]`

```
RouteFinder::RouteFinder ( )
```

Default constructor: creates an empty RouteFinder object.

Returns

Sets data fields appropriately.

2.1.1.2 RouteFinder() [2/3]

```
RouteFinder::RouteFinder (
            const RouteFinder & r )
```

Copy constructor: performs a deep copy of r.

Parameters

r	A RouteFinder object to copy.

Returns

Sets data fields appropriately.

2.1.1.3 RouteFinder() [3/3]

```
RouteFinder::RouteFinder (
            const std::string & f )
```

Parameterized constructor.

Parameters

f	A filename to load data from. See project description for the file format. If f does not exist or does not have the correct format, a RouteFinder object is created based on the input file "sample.txt", which can be assumed to always exist.

Returns

Sets data fields appropriately.

2.1.2 Member Function Documentation

2.1.2.1 getBoxes()

```
std::vector<int> RouteFinder::getBoxes ( ) const
```

Gets the vector containing the number of boxes for the houses.

Returns

The vector representing the integer number of boxes for the houses.

2.1.2.2 getBoxNum()

```
std::vector<int> RouteFinder::getBoxNum ( ) const
```

Gets the total number of boxes on each route.

Returns

A vector of integers showing the total number of boxes on each route.

2.1.2.3 getDist()

```
std::vector< dLinkedList<double> > RouteFinder::getDist ( ) const
```

Gets the vector containing the distance between houses.

Returns

The vector representing the distances.

2.1.2.4 getFinalDist()

```
double RouteFinder::getFinalDist ( ) const
```

Get the total distance of the fleet.

Returns

The total fleet distance per day.

2.1.2.5 getHouseNum()

```
int RouteFinder::getHouseNum ( ) const
```

Gets the total number of houses.

Returns

The integer number representing the total number of houses.

2.1.2.6 getMaxBoxes()

```
int RouteFinder::getMaxBoxes ( ) const
```

Gets the maximum number of boxes a delivery truck can carry.

Returns

An integer representing the maximum number of boxes per truck.

2.1.2.7 getMaxDist()

```
double RouteFinder::getMaxDist ( ) const
```

Gets the the maximum distance for each delivery truck.

Returns

A number representing the maximum distance a delivery truck can travel per day.

2.1.2.8 getRouteDist()

```
std::vector<double> RouteFinder::getRouteDist ( ) const
```

Gets the distance of each of the final routes.

Returns

A vector of real numbers showing the cost for each route.

2.1.2.9 getRoutes()

```
std::vector< dLinkedList<int> > RouteFinder::getRoutes ( ) const
```

Gets the final routes.

Returns

A vector of doubly linked list. See project description.

2.1.2.10 operator=()

```
RouteFinder& RouteFinder::operator= (
             RouteFinder x )
```

Assignment operator overloading: performs a deep copy of x.

Parameters

x	A RouteFinder object to copy.

Returns

A RouteFinder object with identical contents as x.

2.1.2.11 setBoxes()

```
void RouteFinder::setBoxes (
            std::vector< int > b )
```

Sets the number of boxes for the houses.

Parameters

b	is the vector containing the number of boxes to be delivered to each house.

Returns

Set data fields appropriately.

2.1.2.12 setDist()

```
void RouteFinder::setDist (
            std::vector< dLinkedList< double > > d )
```

Sets the distances between houses.

Parameters

d	is the distance matrix in the form of a vector of doubly linked list. (See project description.)

Returns

Set data fields appropriately.

2.1.2.13 setHouseNum()

```
void RouteFinder::setHouseNum (
            int n )
```

Sets the total number of houses.

Parameters

n	is the total number of houses.

Returns

Set data fields appropriately.

2.1.2.14 setMaxBoxes()

```
void RouteFinder::setMaxBoxes (
             int m )
```

Sets the maximum number of boxes a delivery truck can carry.

Parameters

m	is the maximum number of boxes.

Returns

Set data fields appropriately.

2.1.2.15 setMaxDist()

```
void RouteFinder::setMaxDist (
             double d )
```

Sets the the maximum distance for each delivery truck.

Parameters

d	is the maximum distance a truck can travel in a day.

Returns

Set data fields appropriately.

2.1.2.16 solve()

```
void RouteFinder::solve ( )
```

Heuristic algorithm to solve Problem 1 (see project description).

Returns

sets data fields appropriately.

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

- RouteFinder.hpp

Index