CE100 Algorithms and Programming II HW3

v1.0.0

Generated by Doxygen 1.9.6

Chapter 1

ce100 - Homework 3

TEAM MEMBERS

- Nefise GÜLLÜ - 211401024 - nefise_gullu21@erdogan.edu.tr
- Ali Alptuğ DEMİR - 211401005 - alialptug_demir21@erdogan.edu.tr

REQUIRMENTS

- Visual Studio 2022
- Notepad++
- Git Extensions
- Git Bash
- WebSite: []() https://ucoruh.github.io/ce100-algorithms-and-programming-II/

ENVIRONMENT SETUP

- Visual Studio Community Edition
- .Net Core 6.0 Framework
- XUnit
- Choco Package Manager
- ReportGenerator
- Doxygen

RUNNING

Functions run via Visual Studio 2022.

TESTING

Functions unit tested via Visual Studio 2022.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures 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 ce100_hw3_algo_lib_cs Namespace Reference

Data Structures

- class AssemblyGuide

  Represents an assembly guide that determines the order in which items should be assembled based on their dependencies.
- class CityRoadNetwork

  Represents a city road network and provides methods to find the shortest path between two vertices.
- class Edge
- class HuffmanAlgorithm

  Class for performing Huffman encoding and decoding.
- class TreePipelineSystem

  Represents a tree pipeline system that connects multiple trees.

5.2 ce100_hw3_algo_test_cs Namespace Reference

Data Structures

- class AssemblyGuideTests
- class CityRoadNetworkTests
- class HuffmanAlgorithmTests
- class TreePipelineSystemTests

Chapter 6

Data Structure Documentation

6.1 ce100_hw3_algo_lib_cs.AssemblyGuide Class Reference

Represents an assembly guide that determines the order in which items should be assembled based on their dependencies.

Public Member Functions

- AssemblyGuide ()
- void AddItemDependency (string item, List< string > dependencies)

  Adds a dependency for an item.
- void AddItemDescription (string item, string description)

  Adds a description for an item.
- ArrayList GetAssemblySteps ()

  Gets the assembly steps in the correct order.
- void PerformTopologicalSort ()

Private Member Functions

- bool DFS (string item)

  Performs depth-first search (DFS) to check for cycles and determine the order of items.

Private Attributes

- Dictionary< string, List< string > > dependencies
- List< string > assemblyOrder
- HashSet< string > visited
- HashSet< string > recursionStack
- Dictionary< string, string > itemDescriptions

6.1.1 Detailed Description

Represents an assembly guide that determines the order in which items should be assembled based on their dependencies.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 AssemblyGuide()

```
ce100_hw3_algo_lib_cs.AssemblyGuide.AssemblyGuide ( )  [inline]
```

References ce100_hw3_algo_lib_cs.AssemblyGuide.assemblyOrder, ce100_hw3_algo_lib_cs.AssemblyGuide.dependencies, ce100_hw3_algo_lib_cs.AssemblyGuide.itemDescriptions, ce100_hw3_algo_lib_cs.AssemblyGuide.recursionStack, and ce100_hw3_algo_lib_cs.AssemblyGuide.visited.

6.1.3 Member Function Documentation

6.1.3.1 AddItemDependency()

```
void ce100_hw3_algo_lib_cs.AssemblyGuide.AddItemDependency (
            string item,
            List< string > dependencies )  [inline]
```

Adds a dependency for an item.

Parameters

item	The item.
dependencies	The dependencies of the item.

References ce100_hw3_algo_lib_cs.AssemblyGuide.dependencies.

Referenced by ce100_hw3_algo_test_cs.AssemblyGuideTests.TestAssemblyGuide().

Here is the caller graph for this function:

6.1.3.2 AddItemDescription()

```
void ce100_hw3_algo_lib_cs.AssemblyGuide.AddItemDescription (
            string item,
            string description )  [inline]
```

Adds a description for an item.

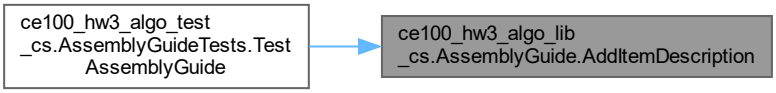
Parameters

item	The item.
description	The description of the item.

References ce100_hw3_algo_lib_cs.AssemblyGuide.itemDescriptions.

Referenced by ce100_hw3_algo_test_cs.AssemblyGuideTests.TestAssemblyGuide().

Here is the caller graph for this function:



6.1.3.3 DFS()

```
bool ce100_hw3_algo_lib_cs.AssemblyGuide.DFS (
            string item )  [inline], [private]
```

Performs depth-first search (DFS) to check for cycles and determine the order of items.

Parameters

item	The current item being visited.

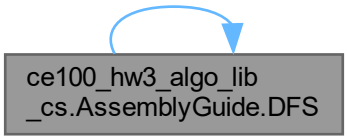
Returns

True if a cycle is detected, false otherwise.

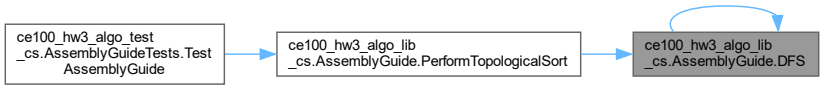
References ce100_hw3_algo_lib_cs.AssemblyGuide.assemblyOrder, ce100_hw3_algo_lib_cs.AssemblyGuide.dependencies, ce100_hw3_algo_lib_cs.AssemblyGuide.DFS(), ce100_hw3_algo_lib_cs.AssemblyGuide.recursionStack, and ce100_hw3_algo_lib_cs.AssemblyGuide.visited.

Referenced by ce100_hw3_algo_lib_cs.AssemblyGuide.DFS(), and ce100_hw3_algo_lib_cs.AssemblyGuide.PerformTopologicalSort

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.4 GetAssemblySteps()

```
ArrayList ce100_hw3_algo_lib_cs.AssemblyGuide.GetAssemblySteps ( )  [inline]
```

Gets the assembly steps in the correct order.

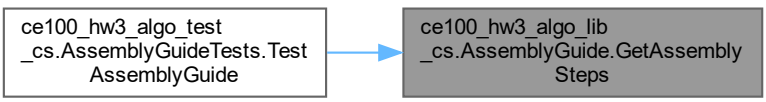
Returns

An ArrayList containing the assembly steps.

References ce100_hw3_algo_lib_cs.AssemblyGuide.assemblyOrder, and ce100_hw3_algo_lib_cs.AssemblyGuide.itemDescriptions.

Referenced by ce100_hw3_algo_test_cs.AssemblyGuideTests.TestAssemblyGuide().

Here is the caller graph for this function:



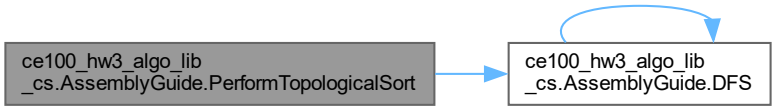
6.1.3.5 PerformTopologicalSort()

```
void ce100_hw3_algo_lib_cs.AssemblyGuide.PerformTopologicalSort ( )  [inline]
```

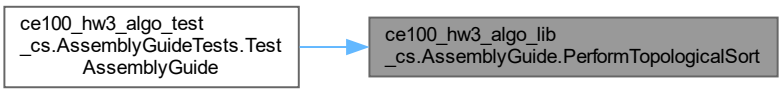
References ce100_hw3_algo_lib_cs.AssemblyGuide.assemblyOrder, ce100_hw3_algo_lib_cs.AssemblyGuide.dependencies, ce100_hw3_algo_lib_cs.AssemblyGuide.DFS(), and ce100_hw3_algo_lib_cs.AssemblyGuide.visited.

Referenced by ce100_hw3_algo_test_cs.AssemblyGuideTests.TestAssemblyGuide().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.4 Field Documentation

6.1.4.1 assemblyOrder

```
List<string> ce100_hw3_algo_lib_cs.AssemblyGuide.assemblyOrder  [private]
```

Referenced by ce100_hw3_algo_lib_cs.AssemblyGuide.AssemblyGuide(), ce100_hw3_algo_lib_cs.AssemblyGuide.DFS(), ce100_hw3_algo_lib_cs.AssemblyGuide.GetAssemblySteps(), and ce100_hw3_algo_lib_cs.AssemblyGuide.PerformTopologicalSort(

6.1.4.2 dependencies

```
Dictionary<string, List<string> > ce100_hw3_algo_lib_cs.AssemblyGuide.dependencies  [private]
```

Referenced by ce100_hw3_algo_lib_cs.AssemblyGuide.AddItemDependency(), ce100_hw3_algo_lib_cs.AssemblyG ce100_hw3_algo_lib_cs.AssemblyGuide.DFS(), and ce100_hw3_algo_lib_cs.AssemblyGuide.PerformTopologicalSort().

6.1.4.3 itemDescriptions

```
Dictionary<string, string> ce100_hw3_algo_lib_cs.AssemblyGuide.itemDescriptions  [private]
```

Referenced by ce100_hw3_algo_lib_cs.AssemblyGuide.AddItemDescription(), ce100_hw3_algo_lib_cs.AssemblyGuide.AssemblyGu and ce100_hw3_algo_lib_cs.AssemblyGuide.GetAssemblySteps().

6.1.4.4 recursionStack

```
HashSet<string> ce100_hw3_algo_lib_cs.AssemblyGuide.recursionStack  [private]
```

Referenced by ce100_hw3_algo_lib_cs.AssemblyGuide.AssemblyGuide(), and ce100_hw3_algo_lib_cs.AssemblyGuide.DFS().

6.1.4.5 visited

```
HashSet<string> ce100_hw3_algo_lib_cs.AssemblyGuide.visited  [private]
```

Referenced by ce100_hw3_algo_lib_cs.AssemblyGuide.AssemblyGuide(), ce100_hw3_algo_lib_cs.AssemblyGuide.DFS(), and ce100_hw3_algo_lib_cs.AssemblyGuide.PerformTopologicalSort().

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/IKEAProductAssemblyGuid

6.2 ce100_hw3_algo_test_cs.AssemblyGuideTests Class Reference

Public Member Functions

- void TestAssemblyGuide ()

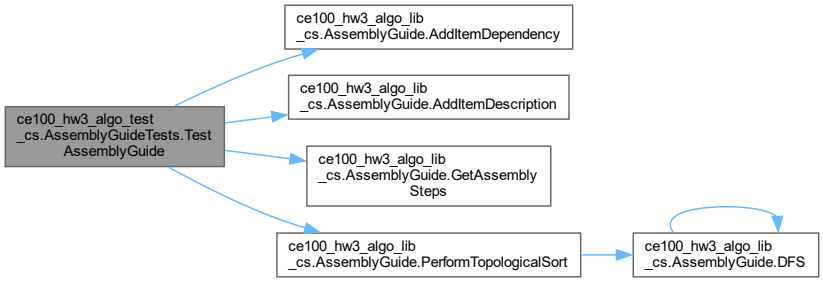
6.2.1 Member Function Documentation

6.2.1.1 TestAssemblyGuide()

```
void ce100_hw3_algo_test_cs.AssemblyGuideTests.TestAssemblyGuide ( )  [inline]
```

References ce100_hw3_algo_lib_cs.AssemblyGuide.AddItemDependency(), ce100_hw3_algo_lib_cs.AssemblyGuide.AddItemDescr
ce100_hw3_algo_lib_cs.AssemblyGuide.GetAssemblySteps(), and ce100_hw3_algo_lib_cs.AssemblyGuide.PerformTopologicalSort(

Here is the call graph for this function:



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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/IKEAProductAssemblyGu

6.3 ce100_hw3_algo_lib_cs.CityRoadNetwork Class Reference

Represents a city road network and provides methods to find the shortest path between two vertices.

Public Member Functions

- CityRoadNetwork (int v)

  Initializes a new instance of the CityRoadNetwork class with the specified number of vertices.
- void AddEdge (int source, int destination, int weight)

  Adds a new edge to the road network.
- List< int > FindShortestPath (int source, int destination)

  Finds the shortest path from a source vertex to a destination vertex in the road network.

Private Attributes

- int vertices
- List< Edge > edges

6.3.1 Detailed Description

Represents a city road network and provides methods to find the shortest path between two vertices.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 CityRoadNetwork()

```
ce100_hw3_algo_lib_cs.CityRoadNetwork.CityRoadNetwork (
            int v )  [inline]
```

Initializes a new instance of the CityRoadNetwork class with the specified number of vertices.

Parameters

v	Number of vertices in the road network.

References ce100_hw3_algo_lib_cs.CityRoadNetwork.edges, and ce100_hw3_algo_lib_cs.CityRoadNetwork.vertices.

6.3.3 Member Function Documentation

6.3.3.1 AddEdge()

```
void ce100_hw3_algo_lib_cs.CityRoadNetwork.AddEdge (
            int source,
            int destination,
            int weight )  [inline]
```

Adds a new edge to the road network.

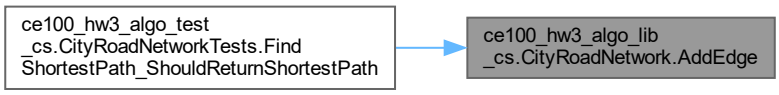
Parameters

source	Source vertex of the edge.
destination	Destination vertex of the edge.
weight	Weight of the edge.

References ce100_hw3_algo_lib_cs.CityRoadNetwork.edges.

Referenced by ce100_hw3_algo_test_cs.CityRoadNetworkTests.FindShortestPath_ShouldReturnShortestPath().

Here is the caller graph for this function:



6.3.3.2 FindShortestPath()

```
List< int > ce100_hw3_algo_lib_cs.CityRoadNetwork.FindShortestPath (
            int source,
            int destination )  [inline]
```

Finds the shortest path from a source vertex to a destination vertex in the road network.

Parameters

source	Source vertex of the path.
destination	Destination vertex of the path.

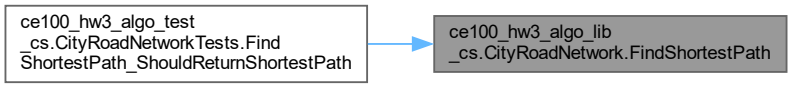
Returns

List of vertices representing the shortest path.

References ce100_hw3_algo_lib_cs.Edge.Destination, ce100_hw3_algo_lib_cs.CityRoadNetwork.edges, ce100_hw3_algo_lib_cs.Ed
ce100_hw3_algo_lib_cs.CityRoadNetwork.vertices, and ce100_hw3_algo_lib_cs.Edge.Weight.

Referenced by ce100_hw3_algo_test_cs.CityRoadNetworkTests.FindShortestPath_ShouldReturnShortestPath().

Here is the caller graph for this function:



6.3.4 Field Documentation

6.3.4.1 edges

```
List<Edge> ce100_hw3_algo_lib_cs.CityRoadNetwork.edges  [private]
```

Referenced by ce100_hw3_algo_lib_cs.CityRoadNetwork.AddEdge(), ce100_hw3_algo_lib_cs.CityRoadNetwork.CityRoadNetwork(),
and ce100_hw3_algo_lib_cs.CityRoadNetwork.FindShortestPath().

6.3.4.2 vertices

```
int ce100_hw3_algo_lib_cs.CityRoadNetwork.vertices  [private]
```

Referenced by ce100_hw3_algo_lib_cs.CityRoadNetwork.CityRoadNetwork(), and ce100_hw3_algo_lib_cs.CityRoadNetwork.FindSh

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/BellmanFord.cs

6.4 ce100_hw3_algo_test_cs.CityRoadNetworkTests Class Reference

Public Member Functions

- void FindShortestPath_ShouldReturnShortestPath ()

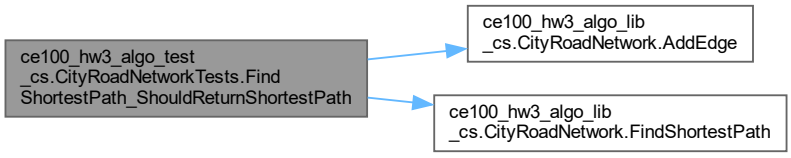
6.4.1 Member Function Documentation

6.4.1.1 FindShortestPath_ShouldReturnShortestPath()

```
void ce100_hw3_algo_test_cs.CityRoadNetworkTests.FindShortestPath_ShouldReturnShortestPath ( )
[inline]
```

References ce100_hw3_algo_lib_cs.CityRoadNetwork.AddEdge(), and ce100_hw3_algo_lib_cs.CityRoadNetwork.FindShortestPath(

Here is the call graph for this function:



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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/BellmanFordTest.cs

6.5 ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T > Class Template Reference

Public Member Functions

- DisjointSet (int size)
- int FindSet (int element)
- bool AreInSameSet (int element1, int element2)
- void MergeSets (int element1, int element2)

Private Attributes

- int[ ] parent
- int[ ] rank

6.5.1 Constructor & Destructor Documentation

6.5.1.1 DisjointSet()

```
ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.DisjointSet (
            int size )  [inline]
```

References ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.parent, and ce100_hw3_algo_lib_cs.TreePipelineSystem.

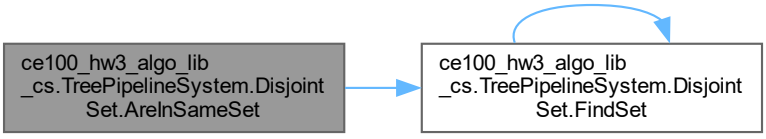
6.5.2 Member Function Documentation

6.5.2.1 AreInSameSet()

```
bool ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.AreInSameSet (
            int element1,
            int element2 )  [inline]
```

References ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.FindSet().

Here is the call graph for this function:



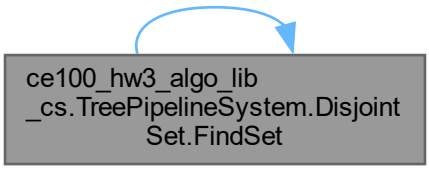
6.5.2.2 FindSet()

```
int ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.FindSet (
            int element )  [inline]
```

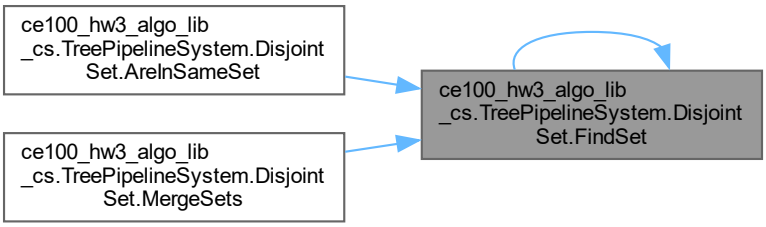
References ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.FindSet(), and ce100_hw3_algo_lib_cs.TreePipelineSyste

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.AreInSameSet(), ce100_hw3_algo_lib_cs.TreePipeline and ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.MergeSets().

Here is the call graph for this function:



Here is the caller graph for this function:

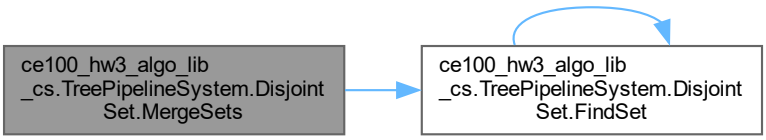


6.5.2.3 MergeSets()

```
void ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.MergeSets (
            int element1,
            int element2 )  [inline]
```

References ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.FindSet(), ce100_hw3_algo_lib_cs.TreePipelineSystem.D and ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.rank.

Here is the call graph for this function:



6.5.3 Field Documentation

6.5.3.1 parent

```
int [] ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.parent  [private]
```

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.DisjointSet(), ce100_hw3_algo_lib_cs.TreePipelineSys
and ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.MergeSets().

6.5.3.2 rank

```
int [] ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.rank  [private]
```

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >.DisjointSet(), and ce100_hw3_algo_lib_cs.TreePipeline

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/TreePipelineSystem.cs

6.6 ce100_hw3_algo_lib_cs.Edge Class Reference

Public Member Functions

- Edge (int source, int destination, int weight)

Properties

- int Source [get, set]
- int Destination [get, set]
- int Weight [get, set]

6.6.1 Constructor & Destructor Documentation

6.6.1.1 Edge()

```
ce100_hw3_algo_lib_cs.Edge.Edge (
            int source,
            int destination,
            int weight )  [inline]
```

References ce100_hw3_algo_lib_cs.Edge.Destination, ce100_hw3_algo_lib_cs.Edge.Source, and ce100_hw3_algo_lib_cs.Edge.We

6.6.2 Property Documentation

6.6.2.1 Destination

```
int ce100_hw3_algo_lib_cs.Edge.Destination  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.Edge.Edge(), and ce100_hw3_algo_lib_cs.CityRoadNetwork.FindShortestPath().

6.6.2.2 Source

```
int ce100_hw3_algo_lib_cs.Edge.Source  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.Edge.Edge(), and ce100_hw3_algo_lib_cs.CityRoadNetwork.FindShortestPath().

6.6.2.3 Weight

```
int ce100_hw3_algo_lib_cs.Edge.Weight  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.Edge.Edge(), and ce100_hw3_algo_lib_cs.CityRoadNetwork.FindShortestPath().

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/BellmanFord.cs

6.7 ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge Class Reference

Public Member Functions

- Edge (int startNode, int endNode, double weight)

Properties

- int StartNode [get]
- int EndNode [get]
- double Weight [get]

6.7.1 Constructor & Destructor Documentation

6.7.1.1 Edge()

```
ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.Edge (
            int startNode,
            int endNode,
            double weight ) [inline]
```

References ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.EndNode, ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.StartNo and ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.Weight.

6.7.2 Property Documentation

6.7.2.1 EndNode

```
int ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.EndNode [get]
```

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.ComputeMST(), and ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.

6.7.2.2 StartNode

```
int ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.StartNode [get]
```

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.ComputeMST(), and ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.

6.7.2.3 Weight

```
double ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.Weight [get]
```

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.Edge().

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/TreePipelineSystem.cs

6.8 ce100_hw3_algo_lib_cs.HuffmanAlgorithm Class Reference

Class for performing Huffman encoding and decoding.

Data Structures

- class HuffmanTree
- class HuffmanTree_mp3
- class Node_mp3
- class Node_Txt

  Node class for text-based Huffman encoding.

Static Public Member Functions

- static void WriteBitArray (BinaryWriter writer, BitArray bits)
- static BitArray ReadBitArray (BinaryReader reader, long byteCount)

6.8.1 Detailed Description

Class for performing Huffman encoding and decoding.

6.8.2 Member Function Documentation

6.8.2.1 ReadBitArray()

```
static BitArray ce100_hw3_algo_lib_cs.HuffmanAlgorithm.ReadBitArray (
            BinaryReader reader,
            long byteCount )  [inline], [static]
```

6.8.2.2 WriteBitArray()

```
static void ce100_hw3_algo_lib_cs.HuffmanAlgorithm.WriteBitArray (
            BinaryWriter writer,
            BitArray bits )  [inline], [static]
```

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/HuffmanAlgorithm.cs

6.9 ce100_hw3_algo_test_cs.HuffmanAlgorithmTests Class Reference

Public Member Functions

- void EncodeAndDecode_Text_Success ()
- void EncodeAndDecode_MP3_Success ()

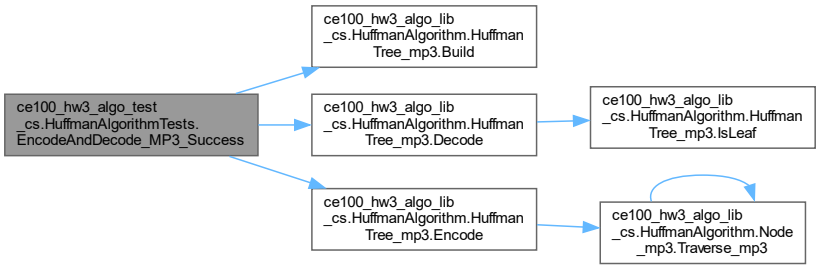
6.9.1 Member Function Documentation

6.9.1.1 EncodeAndDecode_MP3_Success()

```
void ce100_hw3_algo_test_cs.HuffmanAlgorithmTests.EncodeAndDecode_MP3_Success ( )  [inline]
```

References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Build(), ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Huffma
and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Encode().

Here is the call graph for this function:

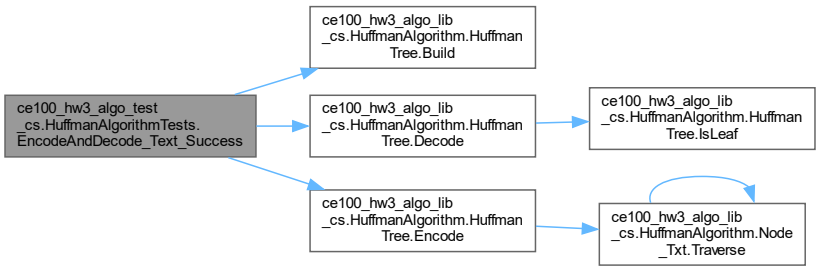


6.9.1.2 EncodeAndDecode_Text_Success()

```
void ce100_hw3_algo_test_cs.HuffmanAlgorithmTests.EncodeAndDecode_Text_Success ( )  [inline]
```

References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Build(), ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree
and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Encode().

Here is the call graph for this function:



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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/HuffmanAlgorithmUnitTes

6.10 ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree Class Reference

Public Member Functions

- void Build (string source)
- BitArray Encode (string source)
- string Decode (BitArray bits)
- bool IsLeaf (Node_Txt node)

Data Fields

- Dictionary< char, int > Frequencies = new Dictionary<char, int>()

Properties

- Node_Txt Root `[get, set]`

Private Attributes

- List< Node_Txt > nodes = new List<Node_Txt>()

6.10.1 Member Function Documentation

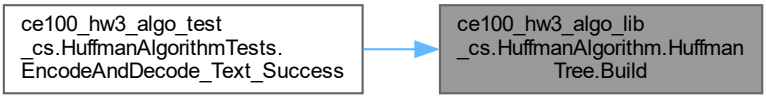
6.10.1.1 Build()

```
void ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Build (
            string source )  [inline]
```

References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Frequencies, ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_
and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.nodes.

Referenced by ce100_hw3_algo_test_cs.HuffmanAlgorithmTests.EncodeAndDecode_Text_Success().

Here is the caller graph for this function:



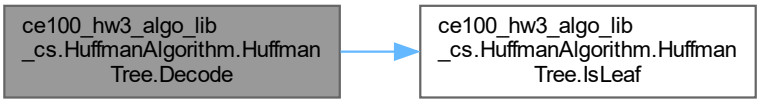
6.10.1.2 Decode()

```
string ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Decode (
            BitArray bits )  [inline]
```

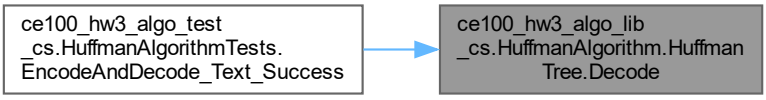
References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.IsLeaf(), ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.L
ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Right, ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Root,
and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Symbol.

Referenced by ce100_hw3_algo_test_cs.HuffmanAlgorithmTests.EncodeAndDecode_Text_Success().

Here is the call graph for this function:



Here is the caller graph for this function:



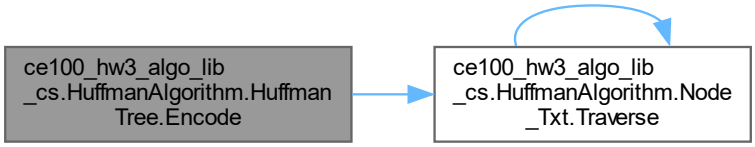
6.10.1.3 Encode()

```
BitArray ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Encode (
            string source )  [inline]
```

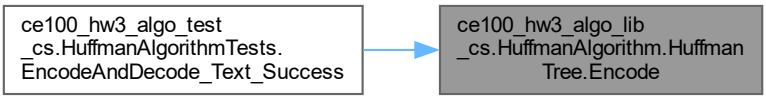
References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Root, and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt

Referenced by ce100_hw3_algo_test_cs.HuffmanAlgorithmTests.EncodeAndDecode_Text_Success().

Here is the call graph for this function:



Here is the caller graph for this function:



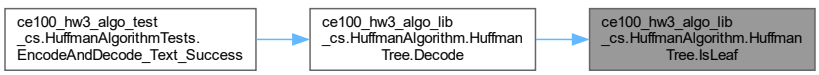
6.10.1.4 IsLeaf()

```
bool ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.IsLeaf (
            Node_Txt node )  [inline]
```

References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Left, and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Rig

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Decode().

Here is the caller graph for this function:



6.10.2 Field Documentation

6.10.2.1 Frequencies

```
Dictionary<char, int> ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Frequencies = new
Dictionary<char, int>()
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Build().

6.10.2.2 nodes

```
List<Node_Txt> ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.nodes = new List<Node_Txt>()
[private]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Build().

6.10.3 Property Documentation

6.10.3.1 Root

```
Node_Txt ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Root  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Decode(), and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Hu

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/HuffmanAlgorithm.cs

6.11 ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3 Class Reference

Public Member Functions

- void Build (byte[ ] source)
- BitArray Encode (byte[ ] source)
- byte[ ] Decode (BitArray bits)
- bool IsLeaf (Node_mp3 node)

Data Fields

- Dictionary< byte, int > Frequencies = new Dictionary<byte, int>()

Properties

- Node_mp3 Root `[get, set]`

Private Attributes

- List< Node_mp3 > nodes = new List<Node_mp3>()

6.11.1 Member Function Documentation

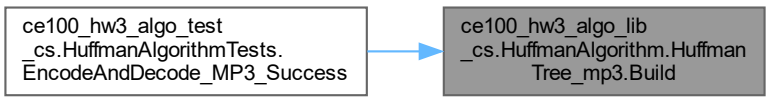
6.11.1.1 Build()

```
void ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Build (
            byte[] source )  [inline]
```

References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Frequencies, ce100_hw3_algo_lib_cs.HuffmanAlgorithm.N and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.nodes.

Referenced by ce100_hw3_algo_test_cs.HuffmanAlgorithmTests.EncodeAndDecode_MP3_Success().

Here is the caller graph for this function:



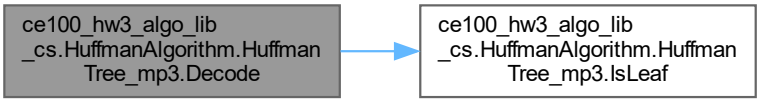
6.11.1.2 Decode()

```
byte[] ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Decode (
            BitArray bits )  [inline]
```

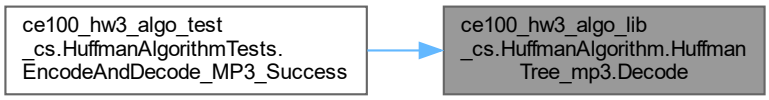
References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.IsLeaf(), and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.H

Referenced by ce100_hw3_algo_test_cs.HuffmanAlgorithmTests.EncodeAndDecode_MP3_Success().

Here is the call graph for this function:



Here is the caller graph for this function:



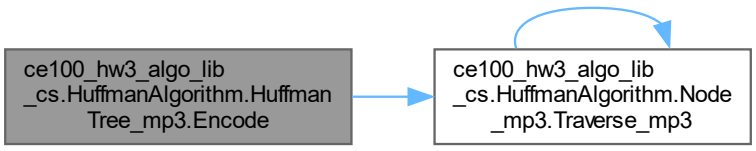
6.11.1.3 Encode()

```
BitArray ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Encode (
            byte[] source )  [inline]
```

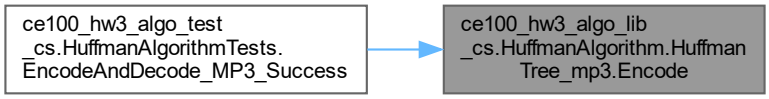
References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Root, and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Nod

Referenced by ce100_hw3_algo_test_cs.HuffmanAlgorithmTests.EncodeAndDecode_MP3_Success().

Here is the call graph for this function:



Here is the caller graph for this function:



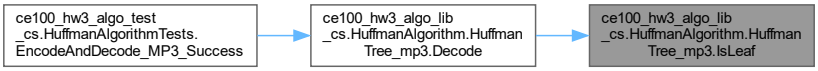
6.11.1.4 IsLeaf()

```
bool ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.IsLeaf (
            Node_mp3 node ) [inline]
```

References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Left, and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.R

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Decode().

Here is the caller graph for this function:



6.11.2 Field Documentation

6.11.2.1 Frequencies

```
Dictionary<byte, int> ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Frequencies = new
Dictionary<byte, int>()
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Build().

6.11.2.2 nodes

```
List<Node_mp3> ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.nodes = new List<Node_mp3>()
[private]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Build().

6.11.3 Property Documentation

6.11.3.1 Root

```
Node_mp3 ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Root  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Decode(), and ce100_hw3_algo_lib_cs.HuffmanAlgorit

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/HuffmanAlgorithm.cs

6.12 ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3 Class Reference

Public Member Functions

- List< bool > Traverse_mp3 (byte? symbol, List< bool > data)

Properties

- byte Symbol `[get, set]`
- int Frequency `[get, set]`
- Node_mp3 Left `[get, set]`
- Node_mp3 Right `[get, set]`

6.12.1 Member Function Documentation

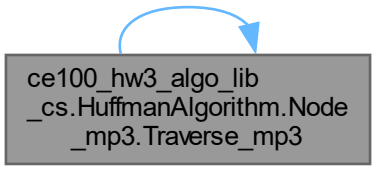
6.12.1.1 Traverse_mp3()

```
List< bool > ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Traverse_mp3 (
            byte?  symbol,
            List< bool > data )  [inline]
```

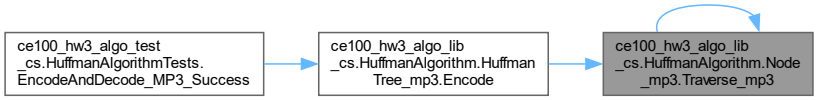
References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Left, ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Right, ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Symbol, and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Traverse_

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Encode(), and ce100_hw3_algo_lib_cs.HuffmanAlgorith

Here is the call graph for this function:



Here is the caller graph for this function:



6.12.2 Property Documentation

6.12.2.1 Frequency

```
int ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Frequency  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.Build().

6.12.2.2 Left

```
Node_mp3 ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Left [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.IsLeaf(), and ce100_hw3_algo_lib_cs.HuffmanAlgorithm

6.12.2.3 Right

```
Node_mp3 ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Right [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3.IsLeaf(), and ce100_hw3_algo_lib_cs.HuffmanAlgorithm

6.12.2.4 Symbol

```
byte ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Symbol [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3.Traverse_mp3().

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/HuffmanAlgorithm.cs

6.13 ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt Class Reference

Node class for text-based Huffman encoding.

Public Member Functions

- List< bool > Traverse (char symbol, List< bool > data)

Properties

- char Symbol [get, set]
- int Frequency [get, set]
- Node_Txt Right [get, set]
- Node_Txt Left [get, set]

6.13.1 Detailed Description

Node class for text-based Huffman encoding.

6.13.2 Member Function Documentation

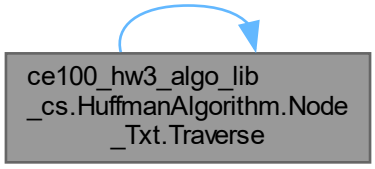
6.13.2.1 Traverse()

```
List< bool > ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Traverse (
            char symbol,
            List< bool > data )  [inline]
```

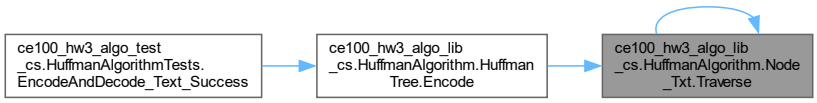
References ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Left, ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Right, and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Traverse().

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Encode(), and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.No

Here is the call graph for this function:



Here is the caller graph for this function:



6.13.3 Property Documentation

6.13.3.1 Frequency

```
int ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Frequency  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Build().

6.13.3.2 Left

```
Node_Txt ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Left  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Decode(), ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Huffma
and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Traverse().

6.13.3.3 Right

```
Node_Txt ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Right  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Decode(), ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Huffma
and ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Traverse().

6.13.3.4 Symbol

```
char ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt.Symbol  [get], [set]
```

Referenced by ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree.Decode().

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/HuffmanAlgorithm.cs

6.14 ce100_hw3_algo_lib_cs.TreePipelineSystem Class Reference

Represents a tree pipeline system that connects multiple trees.

Data Structures

- class DisjointSet
- class Edge

Public Member Functions

- TreePipelineSystem (int numTrees)

  Initializes a new instance of the TreePipelineSystem class with the specified number of trees.
- void ComputeMST ()

  Computes the minimum spanning tree (MST) of the tree pipeline system.
- ArrayList GetMSTEdges ()

  Gets the edges of the minimum spanning tree (MST).

Private Member Functions

- void GenerateRandomTreeLocations ()
- void CalculateDistances ()
- List< Edge > BuildAllEdges ()

Private Attributes

- int numTrees
- double[,] distances
- List< Edge > mstEdges

6.14.1 Detailed Description

Represents a tree pipeline system that connects multiple trees.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 TreePipelineSystem()

```
ce100_hw3_algo_lib_cs.TreePipelineSystem.TreePipelineSystem (
            int numTrees ) [inline]
```

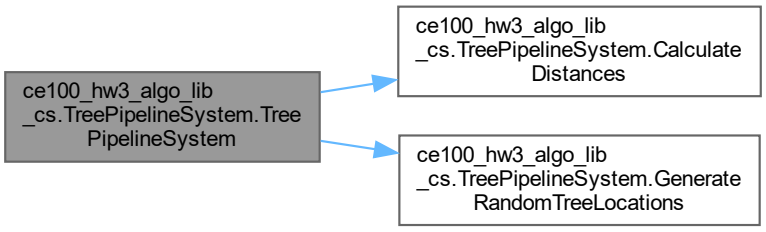
Initializes a new instance of the TreePipelineSystem class with the specified number of trees.

Parameters

numTrees	Number of trees in the system.

References ce100_hw3_algo_lib_cs.TreePipelineSystem.CalculateDistances(), ce100_hw3_algo_lib_cs.TreePipelineSystem.Genera
and ce100_hw3_algo_lib_cs.TreePipelineSystem.numTrees.

Here is the call graph for this function:



6.14.3 Member Function Documentation

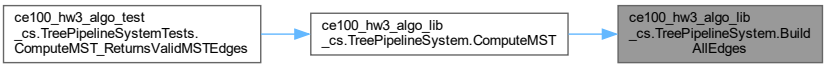
6.14.3.1 BuildAllEdges()

```
List< Edge > ce100_hw3_algo_lib_cs.TreePipelineSystem.BuildAllEdges ( )  [inline], [private]
```

References ce100_hw3_algo_lib_cs.TreePipelineSystem.distances, and ce100_hw3_algo_lib_cs.TreePipelineSystem.numTrees.

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.ComputeMST().

Here is the caller graph for this function:



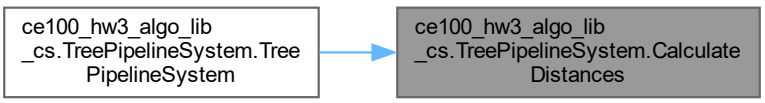
6.14.3.2 CalculateDistances()

```
void ce100_hw3_algo_lib_cs.TreePipelineSystem.CalculateDistances ( )  [inline], [private]
```

References ce100_hw3_algo_lib_cs.TreePipelineSystem.distances, and ce100_hw3_algo_lib_cs.TreePipelineSystem.numTrees.

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.TreePipelineSystem().

Here is the caller graph for this function:



6.14.3.3 ComputeMST()

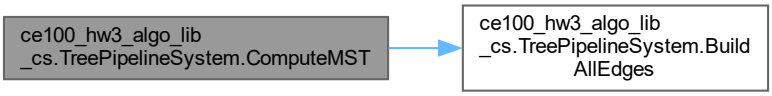
```
void ce100_hw3_algo_lib_cs.TreePipelineSystem.ComputeMST ( )  [inline]
```

Computes the minimum spanning tree (MST) of the tree pipeline system.

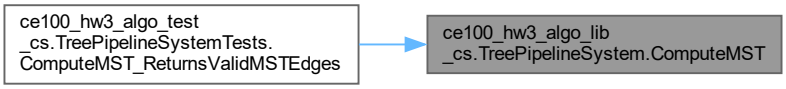
References ce100_hw3_algo_lib_cs.TreePipelineSystem.BuildAllEdges(), ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.EndNo ce100_hw3_algo_lib_cs.TreePipelineSystem.mstEdges, ce100_hw3_algo_lib_cs.TreePipelineSystem.numTrees, and ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge.StartNode.

Referenced by ce100_hw3_algo_test_cs.TreePipelineSystemTests.ComputeMST_ReturnsValidMSTEdges().

Here is the call graph for this function:



Here is the caller graph for this function:



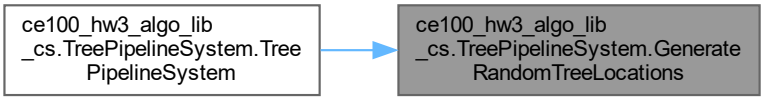
6.14.3.4 GenerateRandomTreeLocations()

```
void ce100_hw3_algo_lib_cs.TreePipelineSystem.GenerateRandomTreeLocations ( )  [inline], [private]
```

References ce100_hw3_algo_lib_cs.TreePipelineSystem.distances, and ce100_hw3_algo_lib_cs.TreePipelineSystem.numTrees.

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.TreePipelineSystem().

Here is the caller graph for this function:



6.14.3.5 GetMSTEdges()

```
ArrayList ce100_hw3_algo_lib_cs.TreePipelineSystem.GetMSTEdges ( )  [inline]
```

Gets the edges of the minimum spanning tree (MST).

Returns

An ArrayList of strings representing the edges of the MST.

References ce100_hw3_algo_lib_cs.TreePipelineSystem.mstEdges.

Referenced by ce100_hw3_algo_test_cs.TreePipelineSystemTests.ComputeMST_ReturnsValidMSTEdges().

Here is the caller graph for this function:

6.14.4 Field Documentation

6.14.4.1 distances

```
double [,] ce100_hw3_algo_lib_cs.TreePipelineSystem.distances  [private]
```

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.BuildAllEdges(), ce100_hw3_algo_lib_cs.TreePipelineSystem.CalculateD
and ce100_hw3_algo_lib_cs.TreePipelineSystem.GenerateRandomTreeLocations().

6.14.4.2 mstEdges

```
List<Edge> ce100_hw3_algo_lib_cs.TreePipelineSystem.mstEdges  [private]
```

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.ComputeMST(), and ce100_hw3_algo_lib_cs.TreePipelineSystem.GetMS

6.14.4.3 numTrees

```
int ce100_hw3_algo_lib_cs.TreePipelineSystem.numTrees  [private]
```

Referenced by ce100_hw3_algo_lib_cs.TreePipelineSystem.BuildAllEdges(), ce100_hw3_algo_lib_cs.TreePipelineSystem.CalculateD ce100_hw3_algo_lib_cs.TreePipelineSystem.ComputeMST(), ce100_hw3_algo_lib_cs.TreePipelineSystem.GenerateRandomTreeLo and ce100_hw3_algo_lib_cs.TreePipelineSystem.TreePipelineSystem().

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/TreePipelineSystem.cs

6.15 ce100_hw3_algo_test_cs.TreePipelineSystemTests Class Reference

Public Member Functions

- void ComputeMST_ReturnsValidMSTEdges ()
- void ComputeMST_ThrowsArgumentException_WhenNumTreesLessThan10 ()

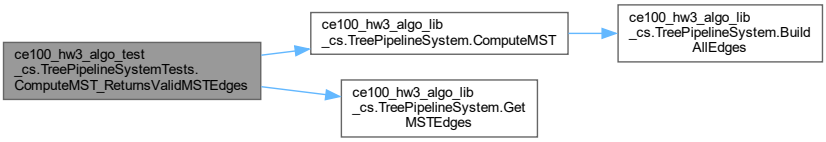
6.15.1 Member Function Documentation

6.15.1.1 ComputeMST_ReturnsValidMSTEdges()

```
void ce100_hw3_algo_test_cs.TreePipelineSystemTests.ComputeMST_ReturnsValidMSTEdges ( )  [inline]
```

References ce100_hw3_algo_lib_cs.TreePipelineSystem.ComputeMST(), and ce100_hw3_algo_lib_cs.TreePipelineSystem.GetMSTE

Here is the call graph for this function:



6.15.1.2 ComputeMST_ThrowsArgumentException_WhenNumTreesLessThan10()

```
void ce100_hw3_algo_test_cs.TreePipelineSystemTests.ComputeMST_ThrowsArgumentException_When↩
NumTreesLessThan10 ( )  [inline]
```

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

- C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/TreePipelineSystemTest.c

Chapter 7

File Documentation

7.1 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/BellmanFord.cs File Reference

Data Structures

- class ce100_hw3_algo_lib_cs.CityRoadNetwork

  Represents a city road network and provides methods to find the shortest path between two vertices.
- class ce100_hw3_algo_lib_cs.Edge

Namespaces

- namespace ce100_hw3_algo_lib_cs

7.2 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/HuffmanAlgorithm.cs File Reference

Data Structures

- class ce100_hw3_algo_lib_cs.HuffmanAlgorithm

  Class for performing Huffman encoding and decoding.
- class ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_Txt

  Node class for text-based Huffman encoding.
- class ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree
- class ce100_hw3_algo_lib_cs.HuffmanAlgorithm.Node_mp3
- class ce100_hw3_algo_lib_cs.HuffmanAlgorithm.HuffmanTree_mp3

Namespaces

- namespace ce100_hw3_algo_lib_cs

7.3 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/IKEAProductAssemblyGuide.cs File Reference

Data Structures

- class ce100_hw3_algo_lib_cs.AssemblyGuide

  Represents an assembly guide that determines the order in which items should be assembled based on their dependencies.

Namespaces

- namespace ce100_hw3_algo_lib_cs

7.4 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/Properties/AssemblyInfo.cs File Reference

7.5 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-lib-cs/TreePipelineSystem.cs File Reference

Data Structures

- class ce100_hw3_algo_lib_cs.TreePipelineSystem

  Represents a tree pipeline system that connects multiple trees.
- class ce100_hw3_algo_lib_cs.TreePipelineSystem.Edge
- class ce100_hw3_algo_lib_cs.TreePipelineSystem.DisjointSet< T >

Namespaces

- namespace ce100_hw3_algo_lib_cs

Variables

- $ Edgefromtree

7.5.1 Variable Documentation

7.5.1.1 Edgefromtree

```
$ Edgefromtree
```

7.6 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/BellmanFordTest.cs File Reference

Data Structures

- class ce100_hw3_algo_test_cs.CityRoadNetworkTests

Namespaces

- namespace ce100_hw3_algo_test_cs

7.7 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/HuffmanAlgorithmUnitTest.cs File Reference

Data Structures

- class ce100_hw3_algo_test_cs.HuffmanAlgorithmTests

Namespaces

- namespace ce100_hw3_algo_test_cs

7.8 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/IKEAProductAssemblyGuideTest.cs File Reference

Data Structures

- class ce100_hw3_algo_test_cs.AssemblyGuideTests

Namespaces

- namespace ce100_hw3_algo_test_cs

7.9 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/TreePipelineSystemTest.cs File Reference

Data Structures

- class ce100_hw3_algo_test_cs.TreePipelineSystemTests

Namespaces

- namespace ce100_hw3_algo_test_cs

7.10 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/ce100-hw3-sln/ce100-hw3-algo-test-cs/Usings.cs File Reference

7.11 C:/Users/Alptuğ/Desktop/Yeni klasor/ce100-hw3-nefise-gullu/README.md File Reference

Index