Table Of Content

Interface FlowAlgorithm

< Methods >

public interface **FlowAlgorithm**

Flow Algorithm Interface

Interface for implementing the flow algorithm.

Author:

Resham Ahluwalia

Version:

1.0

2017-12-01

Methods

findMaxFlow

```
public double findMaxFlow(graph.SimpleGraph graph)
              throws java.lang.Exception
```

Method to calculate the maximum flow in the given simple graph.

Parameters:

graph - Graph in which to calculate max flow.

Returns:

Value of max flow in the given graph.

Throws:

java.lang.Exception - If finding max flow fails

Class FordFulkerson

```
java.lang.Object
    |
    +--algorithms.FordFulkerson
```

All Implemented Interfaces:

FlowAlgorithm

< Constructors > < Methods >

public class **FordFulkerson**
extends java.lang.Object
implements FlowAlgorithm

Ford Fulkerson Algorithm

This class is an implementation of the Ford Fulkerson Method to find a maximum flow in a given network.

Author:

karthik Kolathumani

Version:

1.0

2017-12-01

Constructors

FordFulkerson

```
public  FordFulkerson()
```

Methods

findMaxFlow

```
public double findMaxFlow(graph.SimpleGraph simpleGraph)
```

Calculates the Max Flow

Returns:

double A max Flow Value

Class PreFlowPush

```
java.lang.Object
    |
    +--algorithms.PreFlowPush
```

All Implemented Interfaces:

FlowAlgorithm

< Constructors > < Methods >

---

public class **PreFlowPush**
extends java.lang.Object

implements FlowAlgorithm

Pre Flow Push algorithm

This class implements pre flow push algorithm to find maximum flow in a graph.

Author:

Resham Ahluwalia

Version:

1.0

2017-12-01

Constructors

PreFlowPush

```
public  PreFlowPush()
```

Methods

findMaxFlow

```
public double findMaxFlow(graph.SimpleGraph simpleGraph)
```

Calculates max flow in given graph using pre flow push algorithm.

Parameters:

simpleGraph - Graph in which to find max flow.

Returns:

Value of max flow in the given graph.

Class ScallingFordFulkerson

```
java.lang.Object
    |
    +--algorithms.ScallingFordFulkerson
```

All Implemented Interfaces:

FlowAlgorithm

< Constructors > < Methods >

---

public class **ScallingFordFulkerson**
extends java.lang.Object

implements FlowAlgorithm

The ScallingFordFulkerson program implements the algorithm to improvise the runtime of Ford Fulkerson algorithm by introducing a scaling factor called "delta". This program finds the Maximum flow for a given s-t path of a Network flow Graph.

Author:

Sonal Goswami

Version:

1.0

2017-12-1

Constructors

ScallingFordFulkerson

```
public  ScallingFordFulkerson()
```

Methods

findMaxFlow

```
public double findMaxFlow(graph.SimpleGraph simpleGraph)
```

Calculates max flow in given graph using Scaling Ford Fulkerson algorithm.

Parameters:

simpleGraph - Graph in which we find max flow.

Returns:

Max flow

Class FlowEdge

```
java.lang.Object
    |
    +--model.FlowEdge
```

< Constructors > < Methods >

---

public class **FlowEdge**
extends java.lang.Object

Represents an edge with flow in the flow graph. Edge could be forward or backward edge.

Author:

Resham Ahluwalia

Version:

1.0

2017-12-01

Constructors

FlowEdge

```
public  FlowEdge(FlowVertex origin,
                 FlowVertex dest,
                 double capacity)
```

Creates a flow edge.

Parameters:

origin - Origin vertex of this edge
dest - Destination vertex of this edge
capacity - Capacity of this edge

Methods

getCapacity

```
public double getCapacity()
```

Get capacity of this edge. For backward edges, capacity = flow on corresponding forward edge.

Returns:

Capacity of edge.

---

getDest

```
public FlowVertex getDest()
```

Get destination vertex of this edge.

Returns:

Destination vertex.

---

getFlow

`public double` **`getFlow`**`()`

Get amount of flow on this edge. Flow on backward edges is always zero.

Returns:

Flow on this edge.

---

getName

`public java.lang.String` **`getName`**`()`

Get name of this edge.

Returns:

Name of the edge.

---

getOrigin

`public` `FlowVertex` **`getOrigin`**`()`

Get origin vertex of this edge.

Returns:

Origin vertex.

---

getResidualCapacity

`public double` **`getResidualCapacity`**`()`

Get residual capacity of this edge. For forward edge, residual capacity = capacity - flow. For backward edge, residual capacity = capacity = flow in corresponding forward edge

Returns:

Residual capacity of edge.

---

increaseFlow

```
public void increaseFlow(double increment)
            throws java.lang.Exception
```

Increase flow on the edge by given amount. Also takes care of adding/removing/updating corresponding backward edge if this is forward edge. If this is backward edge, it also updates the flow on corresponding forward edge, and removes this edge if necessary.

Parameters:

increment - Amount by which to increment the flow on this edge

Throws:

java.lang.Exception - If given increment violates capacity constraints

---

isBackwardEdge

```
public boolean isBackwardEdge()
```

Whether the edge is backward edge or forward edge.

Returns:

True if edge is backward edge and false if edge is forward edge.

Class FlowGraph

```
java.lang.Object
    |
    +--model.FlowGraph
```

< Constructors > < Methods >

---

public class **FlowGraph**
extends java.lang.Object

Represents a flow graph.

Author:

Resham Ahluwalia

Version:

1.0

2017-12-01

Constructors

FlowGraph

```
public  FlowGraph(graph.SimpleGraph graph)
        throws java.lang.Exception
```

Creates a flow graph from the simple graph.

Parameters:

graph - Simple graph from which to construct flow graph

Throws:

java.lang.Exception - If graph generation fails.

Methods

addEdge

```
public void addEdge(FlowVertex origin,
                    FlowVertex dest,
                    double capacity)
             throws java.lang.Exception
```

Add edge in the graph.

Parameters:

origin - Origin vertex of edge to add.
dest - Destination vertex of edge to add.
capacity - Capacity of the edge to add.

Throws:

java.lang.Exception - If addition of edge fails

---

addVertex

```
public void addVertex(FlowVertex vertex)
```

Add vertex in the graph.

Parameters:

vertex - Vertex to add in the graph.

---

getSink

```
public FlowVertex getSink()
```

Get sink vertex in the graph.

Returns:

Sink vertex.

getSource

`public FlowVertex getSource()`

Get source vertex in the graph.

Returns:

Source vertex.

getVertex

`public FlowVertex getVertex(java.lang.String name)`

Get vertex of given name in the graph.

Parameters:

name - Name of the vertex to return.

Returns:

Vertex with given name.

getVertices

`public java.util.Collection getVertices()`

Get vertices in the graph.

Returns:

Collection of vertices in the graph.

numberOfEdges

`public int numberOfEdges()`

Get the number of edges in the graph.

Returns:

Number of edges in the graph.

numberOfVertices

`public int numberOfVertices()`

Get the number of vertices in the graph.

Returns:

Number of vertices in the graph.

resetVisited

`public void resetVisited()`

Reset visited flag on all the vertices in the graph.

Class FlowVertex

```
java.lang.Object
    |
    +--model.FlowVertex
```

< Constructors > < Methods >

public class **FlowVertex**
extends java.lang.Object

Represents a vertex in the flow graph.

Author:

Resham Ahluwalia

Version:

1.0

2017-12-01

Constructors

FlowVertex

`public FlowVertex(java.lang.String name)`

Creates a flow vertex.

Parameters:

name - Name of the vertex.

Methods

addEdge

```
public void addEdge(FlowEdge edge)
            throws java.lang.Exception
```

Add given edge to the vertex. Origin of the given edge must be this vertex.

Parameters:

edge - Edge to add to this vertex.

Throws:

java.lang.Exception - If origin of given edge is not this vertex.

---

getEdges

```
public model.FlowEdge[] getEdges()
```

Get outgoing edges from this vertex. This also includes backward edges.

Returns:

Array of outgoing edges from this vertex.

---

getExcess

```
public double getExcess()
```

Get excess of this vertex.

Returns:

Excess of this vertex.

---

getHeight

```
public int getHeight()
```

Get height label of this vertex.

Returns:

Height label of this vertex.

---

getLessHeightNeighborEdge

`public FlowEdge getLessHeightNeighborEdge()`

Get edge outgoing from this vertex such that other end point of that edge has height less than this vertex.

Returns:

Edge if their is neghboring vertex with height less than this vertex, otherwise null.

---

getName

`public java.lang.String getName()`

Get name of this vertex.

Returns:

Name of this vertex.

---

getOutgoingCapacity

`public double getOutgoingCapacity()`

Get total outgoing capacity from this vertex.

Returns:

Total outgoing capacity from this vertex.

---

getOutgoingFlow

`public double getOutgoingFlow()`

Get outgoing flow from this vertex.

Returns:

Total outgoing flow from this vertex.

---

increaseExcess

`public void increaseExcess(double increment)`

Increase excess on this vertex by given amount.

Parameters:

increment - Amount by which to increase excess.

---

incrementHeight

`public void` **`incrementHeight`**`()`

Increment height label of this vertex by 1.

---

isSourceOrSink

`public boolean` **`isSourceOrSink`**`()`

Whether the vertex is source or sink.

Returns:

True if vertex is either source or sink, otherwise false.

---

isVisited

`public boolean` **`isVisited`**`()`

Whether this vertex has been visited or not. Useful in case of graph traversals like DFS.

Returns:

True if vertex has been marked as visited, false otherwise.

---

markVisited

`public void` **`markVisited`**`()`

Mark this vertex as visited.

---

removeEdge

`public void` **`removeEdge`**`(FlowEdge edge)`

Remove given edge from this vertex.

Parameters:

edge - Edge to remove.

---

resetVisited

`public void` **`resetVisited`**`()`

Reset the visited flag on this vertex.

setHeight

```
public void setHeight(int height)
```

Set height label of this vertex.

Parameters:

height - Height of the vertex.

Class tcss543

```
java.lang.Object
    |
    +--tcss543
```

< Constructors > < Methods >

public class **tcss543**
extends java.lang.Object

Maximum Network Flow

Application to find maximum flow in the graph using {@link algorithms.FordFulkerson Ford Fulkerson}, {@link algorithms.ScallingFordFulkerson Scaling Ford Fulkerson} and {@link algorithms.PreFlowPush Pre Flow Push} Algorithm.

Author:

Resham Ahluwalia

Version:

1.0

2017-12-01

Constructors

tcss543

```
public  tcss543()
```

Methods

main

```
public static void main(java.lang.String[] args)
                  throws java.lang.Exception
```

Main method for running application.
Arguments - Expect one argument which is the name of txt file containing input graph.
Output:
Number of vertices and edges in graph.
Max flow output from all three algorithms along with time taken by each algorithm.

Parameters:

args - Program arguments

Throws:

java.lang.Exception - If encounters some error during max flow calculation

INDEX