Laboration 3 i TDA416

Grupp 7: Erik Öhrn, Paula Eriksson Imable

2015-03-06

Filerna som tillhör uppgiften är CompKruskalEdge.java, CompDijkstraPath.java och DirectedGraph.java.

Dessa testas lämpligast genom att starta ShortRoute, förslagsvis med

```
1 ShortRoute SR = new ShortRoute(null);
```

för att använda de filer för noder och bågar som medföljde laborationen.

Dokumentationen finns som javadoc-kommentarer i filerna. De genererade kommentarerna bifogas i detta dokument.

Class DirectedGraph<E extends Edge>

- java.lang.Object
  - DirectedGraph<E>

- Type Parameters:
  `E` - en klass som utökar Edge

```
public class DirectedGraph<E extends Edge>
extends java.lang.Object
```

En klass som kan hitta kortaste vägen mellan två noder samt skapa ett minimalt uppspännande träd.

Author:
Erik Öhrn & Paula Eriksson Imable

- ## Constructor Summary

Constructors

Constructor and Description
`DirectedGraph(int noOfNodes)` Konstruktor, initialerar en ny graf.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`void`	`addEdge(E e)` Lägger till en ny båge, dels i efterföljarlistan och dels i listan som innehåller alla bågar
`java.util.Iterator<E>`	`minimumSpanningTree()` Hittar ett minsta uppspännande träd hos grafen med hjälp av Kruskals algoritm Prioritetskön använder sig av CompKruskalEdge som jämförare.
`java.util.Iterator<E>`	`shortestPath(int from, int to)` Använder en variant av Dijkstras algoritm (en till en) för att hitta den kortaste vägen mellan två noder.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- ## Constructor Detail

- **DirectedGraph**

```
public DirectedGraph(int noOfNodes)
```

Konstruktor, initialerar en ny graf. Grafen använder sig av en efterföljarlisa för att spara bågarna som går från varje nod. Denna använder sig av ett fält av listor, där fältets storlek är antalet noder.

Parameters:
`noOfNodes` - Antalet noder i grafen
Throws:
`java.lang.IndexOutOfBoundsException` - Om antalet noder är mindre än ett

Method Detail

- **addEdge**

```
public void addEdge(E e)
```

Lägger till en ny båge, dels i efterföljarlistan och dels i listan som innehåller alla bågar

Parameters:
`e` - bågen som läggs till
Throws:
`java.lang.NullPointerException` - om e är null

- **shortestPath**

```
public java.util.Iterator<E> shortestPath(int from,
                                          int to)
```

Använder en variant av Dijkstras algoritm (en till en) för att hitta den kortaste vägen mellan två noder. Den sparar de besökta noderna i ett HashSet, mest för att uppslagning av contains() har komplexiteten O(1).

Parameters:
`from` - startnoden
`to` - slutnoden
Returns:
en iterator över vägen från startnoden till slutnoden, eller null om ingen väg hittas
Throws:
`java.lang.IndexOutOfBoundsException` - om from eller to är mindre än noll

- **minimumSpanningTree**

```
public java.util.Iterator<E> minimumSpanningTree()
```

Hittar ett minsta uppspännande träd hos grafen med hjälp av Kruskals algoritm Prioritetskön använder sig av CompKruskalEdge som jämförare. De olika listornas storlekar sparas inte som variabeln då detta efter test tycktes försvåra läsbarheten samtidigt oms .size() ändå på dessa går på O(1) i tidskomplexitet. Kräver en hel sammanbunden graf för att ge ett korrekt svar

Returns:
en iterator över bågarna som utgör trädet

- Detail:
- Field |
- Constr |
- Method

Class CompDijkstraPath<E extends Edge>

- java.lang.Object
- - CompDijkstraPath<E>

- Type Parameters:
  E -

  All Implemented Interfaces:
  java.lang.Comparable<CompDijkstraPath<E>>

```
public class CompDijkstraPath<E extends Edge>
extends java.lang.Object
implements java.lang.Comparable<CompDijkstraPath<E>>
```

Ett köelement som används för Dijkstras algoritm

Author:
Erik Öhrn & Paula Eriksson Imable

- ### Constructor Summary

Constructors

Constructor and Description
CompDijkstraPath(int node, double cost, java.util.ArrayList<E> path) Skapar ett köelement.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
int	compareTo(CompDijkstraPath<E> dijkstraElement) Jämför med ett annat köelement.
double	getCost()
int	getNode()
java.util.ArrayList<E>	getPath()

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ### Constructor Detail

- **CompDijkstraPath**

```
public CompDijkstraPath(int node,
                        double cost,
                        java.util.ArrayList<E> path)
```

Skapar ett köelement. Får en nod, kostnaden dit från startnoden samt vägen dit från startnoden

Parameters:
`node` - noden
`cost` - kostnaden dit från startnoden
`path` - vägen dit från startnoden
Throws:
`java.lang.IndexOutOfBoundsException` - om cost är mindre än noll

Method Detail

- **getPath**

```
public java.util.ArrayList<E> getPath()
```

Returns:
vägen till noden från startnoden

- **getCost**

```
public double getCost()
```

Returns:
kostnaden till noden via vägen den tagit

- **getNode**

```
public int getNode()
```

Returns:
noden

- **compareTo**

```
public int compareTo(CompDijkstraPath<E> dijkstraElement)
```

Jämför med ett annat köelement.

Specified by:
`compareTo` in interface `java.lang.Comparable<CompDijkstraPath<E extends Edge>>`
Parameters:
`dijkstraElement` - det andra elementet
Throws:
`java.lang.NullPointerException` - om det andra elementet är null

- Detail:
- Field |
- Constr |
- Method

Class CompKruskalEdge<E extends Edge>

- java.lang.Object
- - CompKruskalEdge<E>

- All Implemented Interfaces:
  java.util.Comparator<E>

```
public class CompKruskalEdge<E extends Edge>
extends java.lang.Object
implements java.util.Comparator<E>
```

En klass som jämför två bågar för Kruskals algoritm

Author:
Erik Öhrn & Paula Eriksson Imable

- - ## Constructor Summary

  Constructors

  Constructor and Description

  `CompKruskalEdge()`

  - ## Method Summary

    All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
int	compare(E e1, E e2) Compares two edges

    - ### Methods inherited from class java.lang.Object

      equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

    - ### Methods inherited from interface java.util.Comparator

      comparing, comparing, comparingDouble, comparingInt, comparingLong, equals, naturalOrder, nullsFirst, nullsLast, reversed, reverseOrder, thenComparing, thenComparing, thenComparing, thenComparingDouble, thenComparingInt, thenComparingLong

- - ## Constructor Detail

- **CompKruskalEdge**

```
public CompKruskalEdge()
```

Method Detail

- **compare**

```
public int compare(E e1,
                   E e2)
```

Compares two edges

Specified by:
`compare` in interface `java.util.Comparator<E extends Edge>`
Parameters:
`e1` - Instans av en klass som utökar Edge
`e2` - Instans av en klass som utökar Edge
Throws:
`java.lang.NullPointerException` - om något av elementen är null