

<b>ID</b>	1
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Ant
<b>Methoden</b>	calculateLambda(double distance, double pheromone): BigDecimal Berechnung des Lambda-Wertes einer Strecke zwischen zwei Städten
<b>Vorbedingung</b>	-
<b>Eingaben</b>	3, 1
<b>Erwartetess Ergebnis</b>	$\frac{1}{3}$
<b>Ergebnis</b>	$\frac{1}{3}$ <b>Test erfolgreich</b>

<b>ID</b>	2
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Ant
<b>Methoden</b>	calculateLambdas(ArrayList<City> cities): Map<City, BigDecimal> Berechnung der Menge an Lambda-Werten aller Nachbarn
<b>Vorbedingung</b>	City(1), City(2), City(3) (Stadt 2 und 3 haben 3 als Distanz zu Stadt 1)
<b>Eingaben</b>	ArrayList<City>(City(2), City(3))
<b>Erwartetes Ergebnis</b>	Map(Stadt(2), $\frac{1}{3}$ ; Stadt(3), $\frac{1}{3}$ )
<b>Ergebnis</b>	Map(Stadt(2), $\frac{1}{3}$ ; Stadt(3), $\frac{1}{3}$ ) <b>Test erfolgreich</b>

<b>ID</b>	3
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Ant
<b>Methoden</b>	calculateProbability(BigDecimal lambda, BigDecimal sum): BigDecimal Berechnung der Wahrscheinlichkeit, ob eine bestimmte Stadt besucht wird
<b>Vorbedingung</b>	-
<b>Eingaben</b>	$\frac{1}{3}$ , 3
<b>Erwartetes Ergebnis</b>	$\frac{1}{9}$
<b>Ergebnis</b>	$\frac{1}{9}$ <b>Test erfolgreich</b>

<b>ID</b>	4
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Ant
<b>Methoden</b>	calculateProbabilities(ArrayList<City> cities): Map<City, BigDecimal> Berechnung der Wahrscheinlichkeiten aller erreichbaren Nachbarn
<b>Vorbedingung</b>	City(1), City(2), City(3) (Stadt 2 und 3 haben 3 als Distanz zu Stadt 1)
<b>Eingaben</b>	ArrayList<City>(City(2), City(3))
<b>Erwartetes Ergebnis</b>	Map(Stadt(2), 0.5; Stadt(3), 0.5)
<b>Ergebnis</b>	Map(Stadt(2), 0.5; Stadt(3), 0.5) <b>Test erfolgreich</b>