

# Aufgabe 3: Dreiecke zählen

Team: BANDO

Einsendenummer: 00222

1. November 2017

## Inhaltsverzeichnis

1	Lösungsidee	1
2	Umsetzung	1
3	Beispiele	2
4	Quellcode	2

## 1 Lösungsidee

Ein Dreieck ist durch drei Strecken definiert, die sich all gegenseitig in insgesamt drei Eckpunkten schneiden. Möchte man alle Dreiecke in einem Rätselbild finden, so untersucht man alle Trippel an Seiten und überprüft, ob diese ein Dreieck bilden bzw. ob sich diese in genau drei Eckpunkte schneiden.

## 2 Umsetzung

Zunächst wird eine Methode entworfen, die bestimmt, ob zwei Strecken genau einen Schnittpunkt haben. Hierfür werden lineare Funktionen betrachtet, die durch die gegebenen Anfangs- und Endpunkten der Strecken verlaufen. Die Punkte der ersten Strecke werden dabei  $P_1(a_1|b_1)$  und  $P_2(a_2|b_2)$  genannt und die Punkte der zweiten Strecken werden  $Q_1(c_1|d_1)$  und  $Q_2(c_2|d_2)$  genannt. Nun kann für die genannten Funktionen der Form  $p(x) = m_p x + t_p$  und  $q(x) = m_q x + t_q$  mit Hilfe von Schulkenntnissen die entsprechenden Funktionsterme errechnet werden, da jeweils zwei Punkte bekannt sind, durch die die Funktionsgraphen verlaufen:

$$p(x) = m_p x + t_p \quad \text{mit} \quad m_p = \left( \frac{b_1 - b_2}{a_1 - a_2} \right) \quad \text{und} \quad t_p = b_1 - a_1 m_p$$
$$q(x) = m_q x + t_q \quad \text{mit} \quad m_q = \left( \frac{d_1 - d_2}{c_1 - c_2} \right) \quad \text{und} \quad t_q = d_1 - c_1 m_q$$

Setzt man die Funktionsterme gleich, so lässt sich der allgemeine Schnittpunkt folgenderweise errechnen:

$$S(x|y) \quad \text{mit} \quad x = \frac{t_q - t_p}{m_p - m_q} \quad \text{und} \quad y = p(x) \quad \text{bzw.} \quad y = q(x)$$

In zwei Fällen teilt man bei der Herleitung durch Null:

- Wenn zwei Geraden die gleiche Steigung haben und somit parallel zueinander sind. In diesem Fall gibt es nie genau einen Schnittpunkt.
- Wenn mindestens eine Gerade senkrecht zur x-Achse ist. In diesem Fall ist x von  $S(x|y)$  der x-Wert der Senkrechten und  $y = p(x)$  bzw.  $y = q(x)$ . Genau ein Schnittpunkt ist hier genau dann gegeben, wenn die andere Gerade nicht ebenfalls senkrecht zur x-Achse steht.

Existiert genau ein Schnittpunkt, so ist zu überprüfen, ob dieser Schnittpunkt nicht nur auf beiden Funktionsgraphen liegt, sondern auch, ob der Schnittpunkt auf beiden Strecken liegt. hierfür wird überprüft, ob  $x$  zwischen  $a_1$  und  $a_2$  und zwischen  $c_1$  und  $c_2$  liegt. Ebenfalls wird überprüft, ob  $y$  zwischen  $b_1$  und  $b_2$  und zwischen  $d_1$  und  $d_2$  liegt. Ist dies der Fall, so schneiden sich beide Strecken in genau einem Punkt.

### 3 Beispiele

Die Beispiele finden sich mit Visualisierungen im beiliegendem Ordner. Dreieck0.png ist zunächst die Bilddatei, in der alle Strecken eingezeichnet sind. Alle anderen Bilddateien zeigen das gleiche Bild, wobei in jeder Datei jeweils ein Dreieck farbig hervorgehoben wird. Es ist anzumerken, dass im zweiten und fünftem Beispiel scheinbare Dreiecke sind, die das Programm nicht erkannt hat. Betrachtet man die Strecken mit dem Software GeoGebra, so erkennt man unweigerlich, dass diese scheinbaren Dreiecke keine Dreiecke sind, da sich die entsprechenden Strecken nicht schneiden.

### 4 Quellcode

Der folgenden Methoden bestimmen, ob sich zwei Strecken in genau einem Punkt schneiden:

```

1  def max(a, b): # Gibt die groessere von zwei Zahlen a und b zurueck
    if a >= b: # Ueberprueft, ob a groesser oder gleich b ist
3     return a # Gibt a zurueck
    else: # Fuer den Fall, dass a kleiner als b ist:
5     return b # Gibt b zurueck

7  def min(a, b): # Gibt die kleinere von zwei Zahlen a und b zurueck
    if a <= b: # Ueberprueft, ob a kleiner oder gleich b ist
9     return a # Gibt a zurueck
    else: # Fuer den Fall, dass a groesser als b ist:
11    return b # Gibt b zurueck

13 # Ueberprueft, ob eine Zahl x in einem gegebenem Intervall liegt
def imIntervall(x, Intervall):
15     # Ueberprueft, ob x zwischen den Intervallenden liegt
    if Intervall[0] <= x <= Intervall[1]:
17         return True # Gibt Wahr zurueck
    else: # An sonsten
19         return False # Gibt Falsch zurueck

21 # Ueberprueft, ob zwei Strecken genau einen Schnittpunkt haben
def einSchnittpunkt(a1, b1, a2, b2, c1, d1, c2, d2):
23
24     # P1( a1 / b1 )
25     # P2( a2 / b2 )
26     # Q1( c1 / d1 )
27     # Q2( c2 / d2 )

28
29     # Betrachtung der Strecken P1P2 und Q1Q2:
    # p(x) ist die lineare Funktion, auf der P1P2 liegt
31     # q(x) ist die lineare Funktion, auf der Q1Q2 liegt

32
33     Dp = [min(a1, a2), max(a1, a2)] # Definitionsmenge von p
    Wp = [min(b1, b2), max(b1, b2)] # Wertemenge von p
35     Dq = [min(c1, c2), max(c1, c2)] # Definitionsmenge von q
    Wq = [min(d1, d2), max(d1, d2)] # Wertemenge von q
37
38     try:
39         # p(x) = mp * x + tp
        mp = (b1 - b2) / (a1 - a2) # Steigung von p
41         tp = b1 - a1 * mp # Erhoehung von p

42
43     except ZeroDivisionError:
        # => Die Strecke P1P2 steht senkrecht auf der x-Achse

```

```

45
46     try:
47         #  $q(x) = mq \cdot x + tq$ 
48         mq = (d1 - d2) / (c1 - c2) # Steigung von q
49         tq = d1 - c1 * mq # Erhöhung von q
50
51     except ZeroDivisionError:
52         # => Die Strecke Q1Q2 steht senkrecht auf der x-Achse
53         return False # => kein Schnittpunkt
54
55     # Schnittpunkt mit der Gerade, die durch  $x = a1 = a2$  verläuft
56     # und senkrecht auf der x-Achse steht
57     x = a1
58     y = mq * x + tq # y-Wert des Schnittpunkts,  $y = p(x)$ 
59
60     # Überprüfe, ob x in beiden Definitionsbereichen
61     # und y in beiden Wertemengen liegt
62     if imIntervall(x, Dp) and imIntervall(x, Dq)
63     and imIntervall(y, Wp) and imIntervall(y, Wq):
64         # => Es gibt genau einen Schnittpunkt bei (x, y)
65         return [x, y]
66     else:
67         # => Es gibt genau einen Schnittpunkt,
68         # der jedoch nicht auf den Strecken liegt
69         return False
70
71     try:
72         #  $q(x) = mq \cdot x + tq$ 
73         mq = (d1 - d2) / (c1 - c2) # Steigung von q
74         tq = d1 - c1 * mq # Erhöhung von q
75
76     except ZeroDivisionError:
77         # => Die Strecke Q1Q2 steht senkrecht auf der x-Achse
78
79     # Schnittpunkt mit der Gerade, die durch  $x = c1 = c2$  verläuft
80     # und senkrecht auf der x-Achse steht
81     x = c1
82     y = mp * x + tp # y-Wert des Schnittpunkts,  $y = p(x)$ 
83
84     # Überprüfe, ob x in beiden Definitionsbereichen
85     # und y in beiden Wertemengen liegt
86     if imIntervall(x, Dp) and imIntervall(x, Dq)
87     and imIntervall(y, Wp) and imIntervall(y, Wq):
88         # => Es gibt genau einen Schnittpunkt
89         return [x, y]
90     else:
91         # => Es gibt genau einen Schnittpunkt,
92         # der jedoch nicht auf den Strecken liegt
93         return False
94
95     try:
96         x = (tq - tp) / (mp - mq) # x-Koordinate des Schnittpunkts
97         y = mp * x + tp # y-Koordinate des Schnittpunkts
98
99     except ZeroDivisionError:
100         # => Beide Strecken sind parallel zueinander
101         return False # => kein Schnittpunkt

```

```
103 # Ueberpruefe, ob x in beiden Definitionsbereichen
    # und y in beiden Wertemengen liegt
105 if imIntervall(x, Dp) and imIntervall(x, Dq)
    and imIntervall(y, Wp) and imIntervall(y, Wq):
107     return [x, y] # => Es gibt genau einen Schnittpunkt
    else:
109     # => Es gibt genau einen Schnittpunkt,
        # der jedoch nicht auf den Strecken liegt
111     return False
```

---