

## 2. Programmieraufgabe Computerorientierte Mathematik I

Abgabe: 17.11.2023 über den Comajudge bis 17 Uhr

### 1 Problembeschreibung

Es seien  $P = (p_1, p_2)$  und  $Q = (q_1, q_2)$  ganzzahlige Punkte aus  $\mathbb{Z}^2$ . Punkte in  $\mathbb{Z}^2$  werden auch Gitterpunkte (englisch: lattice points) genannt. Wir definieren ein Rechteck

$$R_{(P,Q)} := \left\{ (x_1, x_2) \in \mathbb{R}^2 \mid \min(p_1, q_1) \leq x_1 \leq \max(p_1, q_1), \right. \\ \left. \min(p_2, q_2) \leq x_2 \leq \max(p_2, q_2) \right\},$$

das von den Punkten  $P$  und  $Q$  aufgespannt wird.

Dazu sei ein weiterer ganzzahliger Punkt  $T = (t_1, t_2)$  gegeben, der das Rechteck  $R_T := R_{((0,0),T)}$  definiert. In dieser Programmieraufgabe soll eine Funktion geschrieben werden, die die Anzahl der Gitterpunkte berechnet, die im Rechteck  $R_T \cap R_{(P,Q)}$  liegen, d. h. die Anzahl der Punkte  $X \in \mathbb{Z}^2$  mit  $X \in R_T$  und  $X \in R_{(P,Q)}$ .

Hierbei sind zwei Spezialfälle zu beachten: *degenerierte* und *leere* Rechtecke. Degenerierte Rechtecke haben mindestens eine verschwindende Koordinatendifferenz. Dabei hat die Schnittmenge zwar den Flächeninhalt 0, kann aber dennoch Gitterpunkte enthalten. Für leere Rechtecke gilt  $R_T \cap R_{(P,Q)} = \emptyset$ . Hier ist sowohl die Anzahl der Gitterpunkte, als auch der Flächeninhalt gleich 0.

In Abbildung 1 finden Sie entsprechende Beispiele.

### 2 Aufgabenstellung und Anforderungen

Schreiben Sie eine Funktion

`get_lattice_point_number(P,Q,T),`

die für den Gitterpunkt  $T$  mit positiven Einträgen und den Gitterpunkten  $P$  und  $Q$ , die Anzahl der Gitterpunkte im Rechteck  $R_T \cap R_{(P,Q)}$  berechnet und dementsprechend einen der folgenden Strings mit `return` zurückgibt:

- Ist eine der Koordinaten von  $T$  negativ, dann soll der String

`Die Eingabe ist fehlerhaft.`

zurückgegeben werden.

- Gilt  $R_T \cap R_{(P,Q)} \neq \emptyset$ , dann soll der String  
`Die Anzahl der Gitterpunkte im Rechteck betraegt <L>.`  
zurückgegeben werden, wobei  $L$  die Anzahl der Gitterpunkte in  $R_T \cap R_{(P,Q)}$  ist.
- Gilt  $R_T \cap R_{(P,Q)} = \emptyset$ , dann soll der String  
`Der Schnitt der gegebenen Rechtecke ist leer.`  
zurückgegeben werden.

Im Folgenden werden Funktionen aufgelistet, die von `get_lattice_point_number` als Unterroutrinen aufgerufen werden sollen. Beachten Sie:

**Jede der in 1. bis 3. aufgeführten Funktionen wird vom Comajudge überprüft!**

1. Implementieren Sie eine Funktion

`convert_to_standard(P,Q),`

die zwei ganzzahlige Gitterpunkte  $\tilde{P}, \tilde{Q}$  zurückgibt, sodass  $\tilde{P}$  die linke untere Ecke und  $\tilde{Q}$  die rechte obere Ecke des Rechtecks  $R_{(P,Q)}$  beschreibt. Wir sagen dann, dass das Rechteck  $R_{(\tilde{P},\tilde{Q})}$  in *Standardform* vorliegt.

2. Schreiben Sie eine Funktion

`intersects(P,Q,T),`

die für einen Gitterpunkt  $T$  mit nichtnegativen Einträgen und zwei Gitterpunkten  $P$  und  $Q$ , die ein Rechteck  $R_{(P,Q)}$  in Standardform beschreiben, den boolschen Wert `True` zurückgibt, falls  $R_T \cap R_{(P,Q)} \neq \emptyset$  gilt und andernfalls `False` zurückgibt.

3. Schreiben Sie Funktionen

`get_delta_x1(p1,q1,t1)      und`  
`get_delta_x2(p2,q2,t2),`

welche die  $x_1$ - bzw. die  $x_2$ -Seitenlänge des Rechtecks  $R_T \cap R_{(P,Q)}$  zurückgeben. Hierbei sei  $T = (t_1, t_2)$  und  $R_{(P,Q)}$  mit  $P = (p_1, p_2)$  und  $Q = (q_1, q_2)$  ein Rechteck in Standardform. Diese Funktion soll nur im Fall eines nichtleeren Schnittes aufgerufen werden. (Daher wird sie auch nur für diesen Fall vom Comajudge getestet.)

## 2.1 Eingabe

Die Eingabeparameter  $P, Q, T$  werden den Funktionen als Typ `tuple` übergeben, die jeweils zwei Einträge vom Typ `int` enthalten. Die Eingabeparameter in die Funktionen `get_delta_x*` sind die entsprechenden Einträge aus  $P, Q, T$  vom Typ `int`.

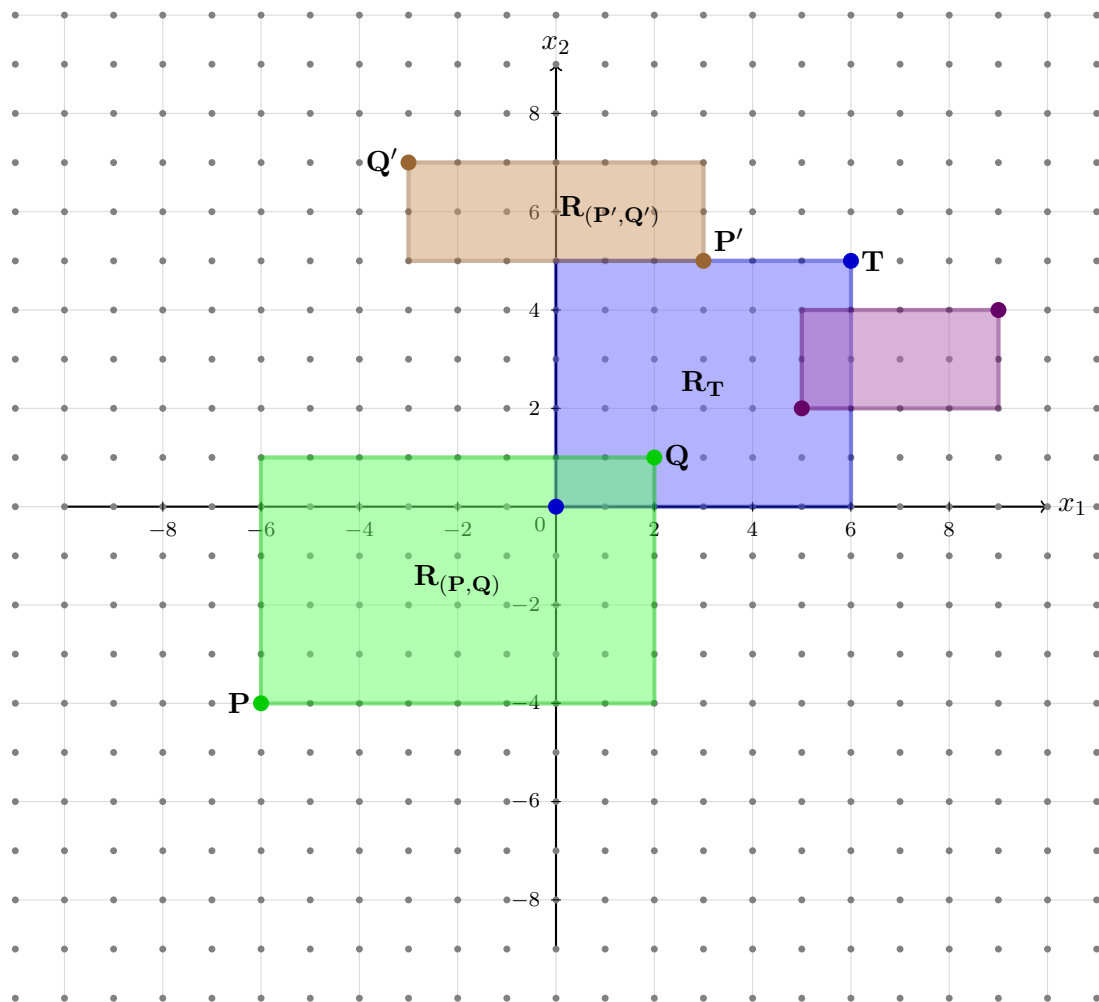


Abbildung 1:  $R_T$  ist gegeben mit  $T = (6, 5)$ . Der Schnitt mit  $R_{(P,Q)}$  enthält 6 Gitterpunkte. Der Schnitt von  $R_{(P',Q')}$  ist degeneriert und enthält 4 Gitterpunkte. Zudem ist das Rechteck  $R_{(P',Q')}$  nicht in Standardform.

## 2.2 Beispielaufufe

Die Rechtecke in den Beispielaufrufen entsprechen Rechtecken aus Abbildung 1.

```
1 python3 -i my_solutionPA02.py
2 >>> P = (-6,-4)
3 >>> Q = (2,1)
4 >>> T = (6,5)
5 >>> answer = get_lattice_point_number(P,Q,T)
6 >>> print(answer)
7 Die Anzahl der Gitterpunkte im Rechteck betraegt 6.

1 python3 -i my_solutionPA02.py
2 >>> P = (3,5)
3 >>> Q = (-3,7)
4 >>> T = (6,5)
5 >>> P,Q = convert_to_standard(P,Q)
6 >>> print(P)
7 (-3,5)
8 >>> print(Q)
9 (3,7)
10 >>> b = intersects(P,Q,T)
11 >>> print(b)
12 True
13 >>> x1 = get_delta_x1(P[0],Q[0],T[0])
14 >>> x2 = get_delta_x2(P[1],Q[1],T[1])
15 >>> print(x1)
16 3
17 >>> print(x2)
18 0
```

## 3 Hinweise

1. Python stellt Funktionen zur Verfügung, die das Minimum (oder Maximum) zurückgeben. Schauen Sie in der Python-Dokumentation nach, was diese Funktionen exakt machen oder implementieren Sie sie selbst.
2. Beachten Sie, dass **tuple** in Python unveränderlich sind, d. h., dass einzelne Einträge in einem Tupel nicht einfach überschrieben werden können. Mit `P[0]` und `P[1]` können Sie den ersten und zweiten Eintrag aus dem Tupel `P` auslesen.
3. Die Ausgabestrings und die Funktionsnamen finden Sie in der Datei `output.txt` auf der ISIS-Seite.
4. Die Rechtecke werden sehr groß. Ihre Programme sollten auch für große Eingaben maximal eine Sekunde benötigen.