2. Programmieraufgabe Computerorientierte Mathematik I

Abgabe: 27.11.2020 über den Comajudge bis 17 Uhr

1 Problembeschreibung

Es seien $a=(a_1,a_2)$ und $b=(b_1,b_2)$ ganzzahlige Punkte aus $\mathbb{Z}^2$. Punkte in $\mathbb{Z}^2$ werden auch Gitterpunkte (englisch: lattice points) genannt. Wir definieren nun ein Rechteck

$$R_{(a,b)} := \left\{(x_1,x_2)\in\mathbb{R}^2 \,:\, \min(a_i,b_i)\leq x_i \leq \max(a_i,b_i) \text{ für } i\in\{1,2\}\right\} \ ,$$

das von den Punkten a und b aufgespannt wird.
Gegeben seien weiter eine nichtnegative ganze Zahl $h\in\mathbb{Z}_{\geq 0}$ und das Rechteck $R_h := R_{(0,(6,h))}$, siehe Skizze. In dieser Programmieraufgabe soll eine Funktion

```
get_lattice_point_number(h,a1,a2,b1,b2) ,
```

geschrieben werden, welche die Anzahl der Punkte $p\in\mathbb{Z}^2$ berechnet, die im Rechteck $R_h\cap R_{(a,b)}$ liegen, für die also gilt $p\in R_h\cap R_{(a,b)}$. Hierbei sind zwei Spezialfälle zu beachten, nämlich *degenerierte* und *leere* Rechtecke, d.h. solche mit verschwindender x- oder y-Koordinatendifferenz und solchen, für die gilt $R_h\cap R_{(a,b)}=\emptyset$. Im ersten Fall hat die Schnittmenge zwar den Flächeninhalt 0, kann aber dennoch Gitterpunkte enthalten. Im zweiten Fall sind sowohl die Zahl der Gitterpunkte, als auch der Flächeninhalt 0.

2 Aufgabenstellung und Anforderungen

Schreiben Sie eine Funktion

```
get_lattice_point_number(h,a1,a2,b1,b2) ,
```

die für nichtnegatives, ganzzahliges $h\in\mathbb{Z}_{\geq 0}$ und ganzzahlige $a=(a_1,a_2)$ und $b=(b_1,b_2)$ aus $\mathbb{Z}^2$ die Zahl der Gitterpunkte in $R_h\cap R_{(a,b)}$ berechnet und dementsprechend einen der folgenden Strings zurückgibt:

- Gilt $R_h\cap R_{(a,b)}\neq\emptyset$, dann soll für $R_h\cap R_{(a,b)}$ der String

  ```
  Die Zahl der Gitterpunkte im resultierenden Rechteck betraegt <L>.
  ```

  zurückgegeben werden, wobei L die Anzahl der Gitterpunkte in $R_h\cap R_{(a,b)}$ ist.
- Gilt $R_h\cap R_{(a,b)}=\emptyset$, dann soll der String

  ```
  Der Schnitt der gegebenen Rechtecke ist leer.
  ```

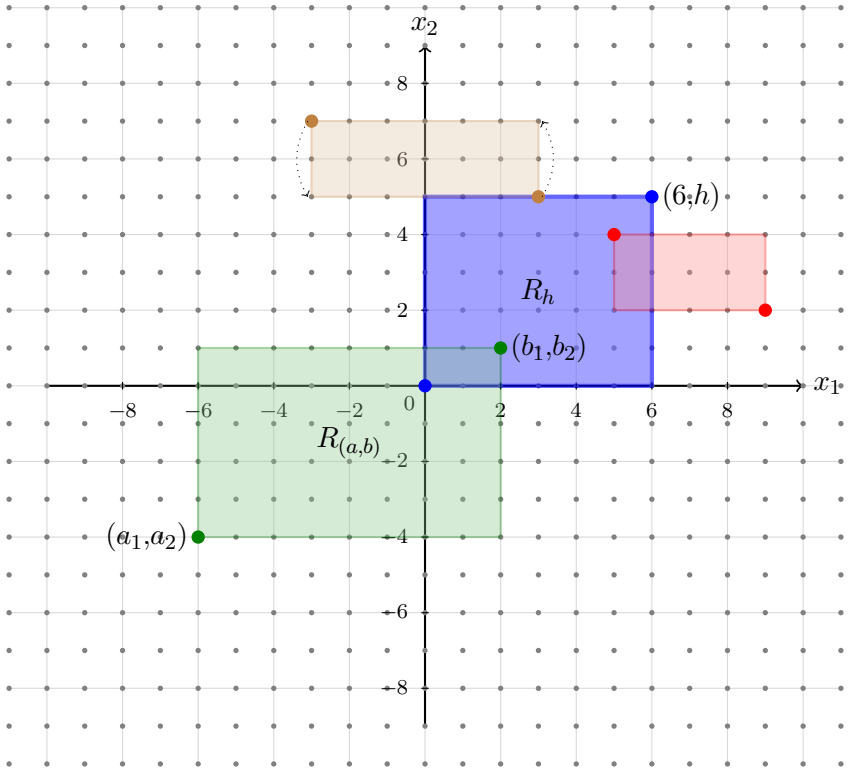
zurückgegeben werden.

- Ist die Eingabe h negativ, dann soll der String

```
Die Eingabe ist fehlerhaft.
```

zurückgegeben werden.

Sie finden diese Strings in der Datei `result_strings.txt` auf der ISIS-Seite.



Im folgenden werden Funktionen aufgelistet, die von `get_lattice_point_number` als Unterroutinen aufgerufen werden sollen.

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

1. Implementieren Sie eine Funktion

```
convert_to_standard(a1,a2,b1,b2)
```

,

die ein ganzzahliges 4-Tupel $(\underline{a}_1,\underline{a}_2,\underline{b}_1,\underline{b}_2)$ zurückgibt, sodass $\underline{a}=(\underline{a}_1,\underline{a}_2)$ die linke untere Ecke und $\underline{b}=(\underline{b}_1,\underline{b}_2)$ die rechte obere Ecke des Rechtecks $R_{(a,b)}$ beschreibt. Wir sagen dann, dass das 4-Tupel $(\underline{a}_1,\underline{a}_2,\underline{b}_1,\underline{b}_2)\in\mathbb{Z}^4$ in *Standardform* vorliegt.

2. Schreiben Sie eine Funktion

   `intersects(h,a1,a2,b1,b2)` ,

   die für jede Eingabe $h \in \mathbb{Z}_{\geq 0}$ und jedes 4-Tupel $(a_1,a_2,b_1,b_2) \in \mathbb{Z}^4$ in Standardform den boolschen Wert `True` zurückgibt, falls $R_h \cap R_{(a,b)} \neq \emptyset$ gilt und andernfals `False` zurückgibt.
3. Schreiben Sie Funktionen

   `get_delta_x1(a1,b1)` und
   `get_delta_x2(h,a2,b2)` ,

   welche die x_1- bzw. die x_2-Seitenlänge des Rechtecks $R_h \cap R_{(a,b)}$ zurückgeben. Hierbei sei wieder (a_1,a_2,b_1,b_2) 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 Funktion `get_lattice_point_number` erhält als Eingabeparameter (in dieser Reihenfolge) die nichtnegative ganze Zahl $h \in \mathbb{Z}_{\geq 0}$ sowie die ganzen Zahlen $a_1,a_2,b_1,b_2 \in \mathbb{Z}$, welche die Punkte $a=(a_1,a_2)$ und $b=(b_1,b_2)$ in $\mathbb{Z}^2$ repräsentieren.

2.2 Rückgabewert

Die Funktion `get_lattice_point_number` soll der Sachlage entsprechend den passenden String aus obiger Liste mit `return` zurückgeben.

2.3 Beispielaufrufe

```
python3 -i my_solutionPA02.py
>>> print(get_lattice_point_number(5,-2,5,0,9))
Die Zahl der Gitterpunkte im resultierenden Rechteck betraegt 1.
>>> a = get_lattice_point_number(5,-2,4,1,9)
>>> print(a)
Die Zahl der Gitterpunkte im resultierenden Rechteck betraegt 4.
>>>
```

3 Tipps und Anmerkungen

1. Überlegen Sie sich die Vorteile dieser Implementierung gegenüber einem Wust aus `if`-Abfragen. (Falls Ihnen langweilig ist - Programmieren Sie diese Aufgabe einmal zum Spass *ohne* jegliche Modularisierung, dann wissen Sie es.) Überlegen Sie sich auch die Schwachpunkte der hier präsentierten Implementierung.
2. Python stellt Funktionen zur Verfügung, "die das Minimum (oder Maximum) zurückgeben". Schauen Sie in der `Python`-Dokumentation nach, was *genau* diese Funktionen machen - oder besser: implementieren Sie sie selbst.