

Junioraufgabe 1: Quadratisch, praktisch, grün

Team-ID: 00000

Team: Constantin

Bearbeiter/-innen dieser Aufgabe:
Constantin

5. September 2024

Inhaltsverzeichnis

Lösungsidee.....	1
Umsetzung.....	2
Beispiele.....	2
Quellcode.....	2

Lösungsidee

Das Grundstück soll in möglichst quadratische Felder aufgeteilt werden, das heißt die Differenz zwischen den Höhe und der Breite eines Feldes sollte möglichst gering sein. Außerdem nimmt die Lösungsidee an, dass alle Felder gleich groß sein müssen und es bevorzugt wird, dass möglichst so viele Felder wie Interessenten entstehen. Die Differenz zwischen der Anzahl an Feldern und Anzahl an Interessenten möglichst gering ist.

Die Lösungsidee basiert auf der mathematischen Idee, dass jede Zahl die keine Primzahl ist in eine Multiplikationsaufgabe zerlegt werden kann. Sollte die Zahl eine Primzahl sein addieren wir 1 und nutzen die [Primzahlücke](#) die besagt, dass die Differenz zweier aufeinanderfolgender Primzahlen mindestens 2 ist. Wir können somit ausschließen, dass wir erneut eine Primzahl erhalten. Die einzige Ausnahme sind die Primzahlen 2 und 3. Dies ist kein Problem, denn wenn die Anzahl an Interessenten kleiner als 10 ist können wir sowieso nicht 1 addieren, da wir höchstens 10% mehr Felder als Interessenten haben dürfen und $x < 10, x + 1 > x * 1,1$. Im Fall, dass die Anzahl der Interessenten kleiner als 10 und eine Primzahl ist, ist die beste Aufteilung immer die Anzahl der Interessenten $\times 1$ wobei jedes Feld die längere Seite (Höhe oder Breite) geteilt durch die Anzahl der Interessenten \times die kürzere Seite entspricht. Jetzt wo wir sicher sind, dass wir unsere Anzahl an Feldern in die wir aufteilen wollen keine Primzahl ist können wir die optimale Aufteilung berechnen. Wir berechnen die optimale Anzahl an horizontalen Aufteilungen mit der Formel $\sqrt{\frac{\text{Breite} \times \text{Interessenten}}{\text{Höhe}}}$. Wir runden dies auf den nächsten Integer, denn wir können das Grundstück nicht in z.B. 3,46 Felder aufteilen. Dann suchen wir den nächsten möglichen Teiler der Anzahl der

Interessenten. Im Fall von 3,46 würden wir also auf 3 abrunden und der nächste Teiler von sagen wir 24 ist 4. Das ist also unsere optimale horizontale Aufteilung. Danach können wir die vertikale Aufteilung sowie die individuelle Feldgröße berechnen.

Umsetzung

Die Datei wird mit

with open(filename, 'r') as file:

lines = file.readlines()

eingelassen und die Werte werden in den Variablen Interessenten, Höhe und Breite gespeichert.

Als nächste überprüfen wir ob Interessenten eine Primzahl ist, dazu nutzen wir Trial Division wie [hier](#) beschrieben. Wenn Interessenten eine Primzahl ist und unter 10 gibt es zwei Möglichkeiten. Ist die *Höhe* > *Breite* ist es besser, das Grundstück horizontal zu teilen, andernfalls ist es besser vertikal zu teilen. Wir geben als beste Aufteilung also entweder

Horizontale Aufteilungen: *Interessenten*, **vertikale Aufteilungen:** 1, **beste Feldhöhe:** *Höhe* / *Interessenten*, **beste Feldbreite:** *Breite*

oder

Horizontale Aufteilungen: 1, **vertikale Aufteilungen:** *Interessenten*, **beste Feldhöhe:** *Höhe*, **beste Feldbreite:** *Breite* / *Interessenten*

zurück. Sollte Interessenten eine Primzahl sein, 10 jedoch nicht unterschreiten addieren wir 1.

Danach berechnen wir die mathematisch beste Anzahl an horizontalen Aufteilungen mit der Formel $\sqrt{\frac{\text{Breite} * \text{Interessenten}}{\text{Höhe}}}$, implementiert sieht dies folgendermaßen aus:

*optimale_horizontale_Aufteilungen = math.sqrt((Breite*Interessente)/Höhe)*

Wir nutzen hier die *sqrt()* Methode aus dem *math* Modul da Python $x^{\frac{1}{2}}$ nicht unterstützt. Danach runden wir mit der *round()* Methode auf den nächsten Integer. Dieser Wert wird in *gerundet_optimale_horizontale_Aufteilungen* gespeichert. Wir überprüfen jetzt mit dem *Modulo (%) Operator* ob unser gerundeter Wert bereits ein Teiler von Interessenten ist.

if Interessenten % gerundet_optimale_horizontale_Aufteilungen == 0

Wenn ja ist das unsere optimale Anzahl an horizontalen Aufteilungen und wir können unser Ergebnis zurückgeben:

Horizontale Aufteilungen: *gerundet_optimale_horizontale_Aufteilungen*, **vertikale Aufteilungen:** *Interessenten* / *gerundet_optimale_horizontale_Aufteilungen*, **beste Feldhöhe:** *Höhe* / *gerundet_optimale_horizontale_Aufteilungen*, **beste Feldbreite:** *Breite* * *gerundet_optimale_horizontale_Aufteilungen* / *Interessenten*

Wenn nein finden wir den Teiler von Interessenten der am nächsten an unseren gerundeten Wert herankommt. Wir berechnen den nächst niedrigeren Teiler mit

```
divisors = [i for i in range(1, int(Interessenten**0.5) + 1) if Interessenten % i == 0]
```

```
divisors += [Interessenten // i for i in divisors if i != Interessenten // i]
```

```
closest_divisor = min(divisors, key=lambda d: abs(d -
gerundet_optimale_horizontale_Aufteilungen))
```

Dieser Wert wird in der Variable *nearest_optimale_horizontale_Aufteilungen* gespeichert. Jetzt wo wir unseren optimalen Teiler gefunden haben können wir mit der selben Formeln unsere restlichen Werte berechnen und zurückgeben, also:

Horizontale Aufteilungen: *nearest_optimale_horizontale_Aufteilungen*, **vertikale Aufteilungen:** *Interessenten / nearest_optimale_horizontale_Aufteilungen*, **beste Feldhöhe:** *Höhe / nearest_optimale_horizontale_Aufteilungen*, **beste Feldbreite:** *Breite * nearest_optimale_horizontale_Aufteilungen / Interessenten*

Beispiele

Garten0.txt:

Interessenten: 23

Höhe: 42

Breite: 66

Beste Aufteilung ist 6 horizontale Felder mal 4 vertikale Felder à 7.0m hoch mal 16.5m breit.

Wir können sehen, dass das Programm in diesem Fall die Anzahl der Felder von 23 auf 24 erhöht, da 23 eine Primzahl ist. Die optimale Anzahl an horizontalen Aufteilungen wird mit

$\sqrt{\frac{66 * 24}{42}} = 6,14$ berechnet, auf 6 gerundet und ist ein Teiler von 24. Dann berechnen wir die restlichen Werte.

Garten1.txt:

Interessenten: 19

Höhe: 15

Breite: 12

Beste Aufteilung ist 4 horizontale Felder mal 5 vertikale Felder à 3.75m hoch mal 2.4m breit.

Dieses Beispiel funktioniert genauso wie Beispiel 1. 19 ist eine Primzahl und wird addieren 1.

$\sqrt{\frac{12 * 20}{15}} = 3,9$, wir runden auf 4, ist ein Teiler von 20 und wir berechnen die restlichen Werte.

Garten2.txt:

Interessenten: 36

Höhe: 55

Breite: 77

Beste Aufteilung ist 6 horizontale Felder mal 6 vertikale Felder à 9.167m hoch mal 12.833m breit.

36 ist keine Primzahl und wir wissen, dass wir genau 36 Felder erhalten können. Die optimale

Anzahl horizontaler Aufteilungen ist $\sqrt{\frac{77 * 36}{55}} = 7.1$. Da 7 kein Teiler von 36 ist finden wir die nächsten kleineren und größeren Teiler also 6 und 9. Die Differenz zwischen 6 und 7 ist kleiner als zwischen 9 und 7 und wir wählen deshalb 6 als optimalen Teiler. Darauf hin berechnen wir die restlichen Werte und geben diese aus.

Garten3.txt:

Interessenten: 101

Höhe: 15

Breite: 15

Beste Aufteilung ist 6 horizontale Felder mal 17 vertikale Felder à 2.5m hoch mal 0.882m breit.

Dieses Beispiel hat den selben Ansatz wie die Beispiele 1 und 2.

101 ist eine Primzahl, 1 addieren.

$\sqrt{\frac{15 * 102}{15}} = 10,1$ wird auf 10 gerundet. Nächster Teiler sind 6 und 17. 6 ist näher an 10 und wir nutzen dies um die restlichen Werte zu berechnen.

Garten4.txt

Interessenten: 1200

Höhe: 37

Breite: 2000

Beste Aufteilung ist 240 horizontale Felder mal 5 vertikale Felder à 0.154m hoch mal 400.0m breit.

1200 ist keine Primzahl. Wir können also direkt nach einem Teiler suchen.

$\sqrt{\frac{2000 * 1200}{37}} = 254,69$, gerundet 255. Der nächsten Teiler sind 240 und 300. 240 ist näher an 255 als 300 und damit der optimale Teiler. Wir berechnen die restlichen Werte und geben diese aus.

Garten5.txt:

Interessenten: 35000

Höhe: 365

Breite: 937

Beste Aufteilung ist 280 horizontale Felder mal 125 vertikale Felder à 1.304m hoch mal 7.496m breit.

35000 ist keine Primzahl. $\sqrt{\frac{937 * 35000}{365}} = 299,75$, gerundet 300. Die nächsten Teiler sind 280 und 350. 280 ist näher an 300 als 350 ist also der optimale Teiler. Wir berechnen die restlichen Werte und gebe diese aus.

Garten6.txt (eigenes Beispiel):

Interessenten: 7

Höhe: 200

Breite: 300

Beste Aufteilung ist 1 horizontale Felder mal 7 vertikale Felder à 200m hoch mal 42.857m breit.

In diesem Beispiel ist unsere Anzahl an Interessenten eine Primzahl, da 7 unter 10 ist und wir deshalb nicht einfach 1 addieren können hat das Programm einen extra check. Die optimale Aufteilung ist in diesem Fall immer 1x7, da die Höhe kleiner ist als die Breite, anders herum hätten wir 7x1. Die beste Feldhöhe und -breite wird trotzdem gleich berechnet.

Quellcode

```
import math

def read_file(filename):
    with open(filename, 'r') as file:
        lines = file.readlines()

        Interessenten = int(lines[0].strip())
        Höhe = int(lines[1].strip())
        Breite = int(lines[2].strip())

    return Interessenten, Höhe, Breite
```

```
def prime_check(number):
    f = 2

    while f * f <= number:
        if number%f == 0:
            return False
        f += 1
    return True

def nearest_multiple_finder(gerundet_optimale_horizontale_Aufteilungen, Interessenten):
    divisors = [i for i in range(1, int(Interessenten**0.5) + 1) if Interessenten % i == 0]
    divisors += [Interessenten // i for i in divisors if i != Interessenten // i]

    closest_divisor = min(divisors, key=lambda d: abs(d -
gerundet_optimale_horizontale_Aufteilungen))

    return closest_divisor

def solve_QPG(Höhe, Breite, Interessenten):
    if prime_check(Interessenten):
        if Interessenten < 10:
            if Höhe > Breite:
                beste_feld_höhe = round(Höhe / Interessenten, 3)
                horizontale_felder_anzahl = Interessenten

                beste_feld_breite = round(Breite, 3)
                vertikale_felder_anzahl = 1

            else:
                beste_feld_höhe = round(Höhe, 3)
                horizontale_felder_anzahl = 1

                beste_feld_breite = round(Breite/Interessenten, 3)
                vertikale_felder_anzahl = Interessenten

            return f"Beste Aufteilung ist {int(horizontale_felder_anzahl)} horizontale Felder mal
{int(vertikale_felder_anzahl)} vertikale Felder à {beste_feld_höhe}m hoch mal {beste_feld_breite}
m breit."
        else:
            Interessenten += 1

    optimale_horizontale_Aufteilungen = math.sqrt((Breite*Interessenten)/Höhe)
    gerundet_optimale_horizontale_Aufteilungen = round(optimale_horizontale_Aufteilungen, 0)

    if Interessenten % gerundet_optimale_horizontale_Aufteilungen == 0:
        beste_feld_höhe = round(Höhe/gerundet_optimale_horizontale_Aufteilungen, 3)
```

```

    beste_feld_breite = round(Breite * gerundet_optimale_horizontale_Aufteilungen/Interessenten,
3)

```

```

    return f"Beste Aufteilung ist {int(gerundet_optimale_horizontale_Aufteilungen)} horizontale
Felder mal {int(Interessenten/gerundet_optimale_horizontale_Aufteilungen)} vertikale Felder à
{beste_feld_höhe}m hoch mal {beste_feld_breite}m breit."

```

```

else:

```

```

    nearest_optimale_horizontale_Aufteilungen =
nearest_multiple_finder(gerundet_optimale_horizontale_Aufteilungen, Interessenten)

```

```

    beste_feld_höhe = round(Höhe/nearest_optimale_horizontale_Aufteilungen, 3)
    beste_feld_breite = round(Breite * nearest_optimale_horizontale_Aufteilungen/Interessenten,
3)

```

```

    return f"Beste Aufteilung ist {int(nearest_optimale_horizontale_Aufteilungen)} horizontale
Felder mal {int(Interessenten/nearest_optimale_horizontale_Aufteilungen)} vertikale Felder à
{beste_feld_höhe}m hoch mal {beste_feld_breite}m breit."

```

```

filename = 'garten2.txt'

```

```

Interessenten, Höhe, Breite = read_file(filename)

```

```

print(solve_QPG(Höhe, Breite, Interessenten))

```