

Problem des Monats · März 2021 LÖSUNG

Hinweise für den Zirkelleiter/die -leiterin:

Geben Sie zunächst nur die offene Problemstellung „Finde Zahlen, deren Summe 20 und deren Produkt möglichst groß ist“ vor, und beobachten Sie, wie die SuS dieses Problem angehen. Die vorgeschlagenen Schritte können als Strukturierungshilfe dienen und zum Weiterdenken motivieren.

Mögliche Lösungen

Schritt 1: *Bilde alle möglichen Paare aus natürlichen Zahlen, deren Summe 20 ist. Welches Zahlenpaar hat das größte Produkt?*

Zahl 1	1	2	3	4	5	6	7	8	9	10
Zahl 2	19	18	17	16	15	14	13	12	11	10
Produkt	19	36	51	64	75	84	91	96	99	100

Schritt 2: *Finde heraus, welche natürlichen Zahlen das größte Produkt haben, wenn die Summe 20 aus drei Zahlen, vier Zahlen usw. gebildet wird. Erkennst du ein Muster?*

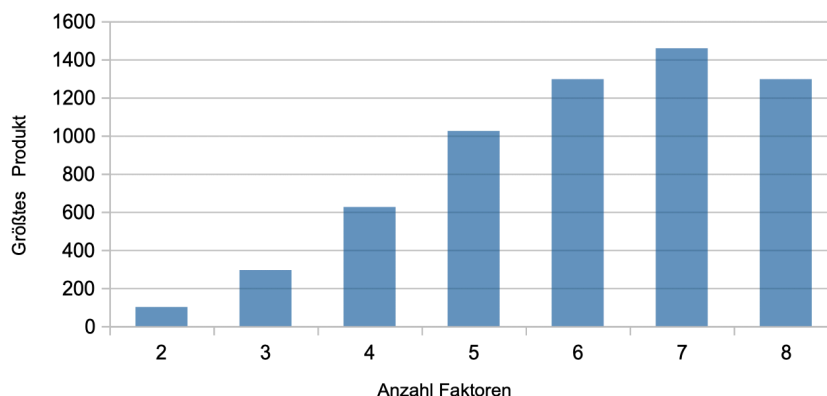
Hier ist eine vollständige tabellarische Lösung zu umfangreich, aber durch Probieren und Überlegen wird man feststellen, dass die Wahl 6, 7, 7 für drei Zahlen zum größten Produkt (294) führt. Bei vier Zahlen erreicht man das Maximum mit 5, 5, 5, 5, nämlich 625. Es liegen folgende Vermutungen nahe:

1. Je mehr Zahlen, desto größer kann das Produkt werden.
2. Je näher die Zahlen beieinander liegen, desto größer wird das Produkt.

Schritt 3: *Bestimme das größtmögliche Produkt aus natürlichen Zahlen, deren Summe 20 ist. (Aus wie vielen Faktoren besteht es?)*

Anzahl Faktoren	2	3	4	5	6	7	8
Faktoren	10,10	6, 7, 7	5, 5, 5, 5	4,4,4,4,4	3,3,3,3,4,4	2,3,3,3,3,3,3	2,2,2,2,3,3,3,3
größtes Produkt	100	294	625	1024	1296	1458	1296

größtes Produkt bei n Faktoren



Schritt 4: Verallgemeinere das Problem für positive rationale Zahlen.

(Tipp: Erstelle einen Funktionsterm für den Zusammenhang zwischen der Anzahl der Faktoren x und dem größtmöglichen Produkt.)

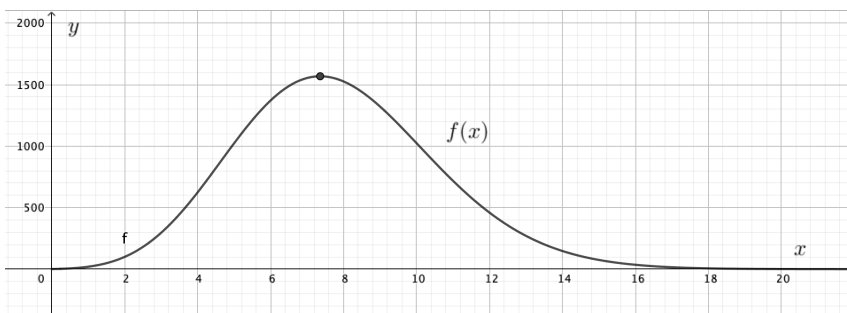
Teilt man die Zahl 20 in x gleich große Faktoren, dann ist jeder Faktor $\frac{20}{x}$. Das Produkt dieser Faktoren ist $\left(\frac{20}{x}\right)^x$. Wir suchen also eine Zahl x , die diesen Term maximiert. Das Säulendiagramm deutet an, dass dieses x in der Nähe von 7 liegen sollte, dass also rund 7 gleich große Faktoren zum größten Produkt führen müssten.

Der Graph der Funktion $f(x) = \left(\frac{20}{x}\right)^x$ ist in der Grafik dargestellt. Der Hochpunkt liegt bei **(7,36 | 1568,05)**.

Wie in Schritt 3 erkennt man, dass eine Zunahme der Faktoren zunächst zu einem Anstieg des maximalen Produkts führt, es aber bei zu vielen Faktoren wieder kleiner wird.

Man kommt also über 1568,05 nicht hinaus. Dies ist das größte Produkt, wenn man die

Zahl in 7 oder 8 Faktoren zerlegt. Da die Funktion um den HP nicht symmetrisch ist, müsste man die Werte 7 und 8 einsetzen. Man erkennt, dass $\left(\frac{20}{7}\right)^7 > \left(\frac{20}{8}\right)^8$ ist und demnach die Lösung 7 Faktoren der Größe $\frac{20}{7}$ ist. Mit Hilfe der Funktion kann gezeigt werden, dass es nur einen Hochpunkt gibt.



Es lassen sich die folgenden Schritte anschließen:

Schritt 5: Untersuche das Problem auch mit negativen Zahlen.

Mit negativen Zahlen kann das Produkt unendlich groß werden. Man könnte z.B. 3000, -2979 und -1 wählen. Dann wäre das Produkt 8.937.000. Hier sind dann keine Grenzen mehr gesetzt.

Schritt 6: Berechne das Maximum für positive Zahlen mit den Methoden der Analysis.

Um die Ableitung zu bilden, wendet man z.B. die Quotientenregel auf $f(x) = \frac{20^x}{x^x}$ an. Den Term x^x leitet man über die Beziehung $x^x = e^{x \ln(x)}$ zu $(x^x)' = (\ln x + 1)x^x$ ab. Daraus folgt dann $f'(x) = (\ln 20 - \ln x - 1) \left(\frac{20}{x}\right)^x$. Die Nullstelle dieser Funktion liegt bei $x = \frac{20}{e} \approx 7,3576$. Interessanterweise wird das Produkt dann maximal, wenn jeder Faktor die Eulersche Zahl ist.

Schritt 7: Man könnte noch nachweisen, dass das Produkt gleicher Faktoren immer größer ist als das Produkt ungleicher Faktoren. Das wäre eine sehr vertiefende Forscherfrage, da es nicht trivial ist, dies zu zeigen.

Schritt 8: Entwickle selbst ein ähnliches Problem.

Mathematikzirkel

Schritt 9: Beim Testen des Problems kam ein Schüler auf die Idee, ein Computerprogramm dazu zu schreiben. Hier ist eine Variante von Niklas Buhr in Python:

```
# Skript zur Lösung des Problems des Monats März 2021 (mit Kommentaren)
# written by Niklas Buhr
```

```
from random import randint # Importieren der Funktion random.randint()
```

```
# Definieren einer Funktion, die alle Werte einer Liste addiert
```

```
def list_sum(elements):
    _sum = 0
    for i in elements:
        _sum += i
    return _sum
```

```
# Definieren einer Funktion, die alle Werte einer Liste multipliziert
```

```
def list_product(elements):
    product = 1
    for i in elements:
        product *= i
    return product
```

```
# Definieren einer Funktion, die eine Liste mit zufälligen Werten erzeugt, welche addiert 20 ergeben
```

```
def create_random_list():
    random_list = []
    while True:
        if list_sum(random_list) == 20:
            break
        elif list_sum(random_list) < 20:
            random_int = randint(1.0, 10.0)
            random_list.append(random_int)
        else:
            random_list = []
    return random_list
```

```
# Definieren der Hauptfunktion
```

```
def main():
    runs = 0
    best_product = 0
    best_list = []
    while True:
        runs += 1 # Eine Variable, die vorhanden ist, um herauszufinden, wie oft diese Schleife schon gelaufen
ist
        current_list = create_random_list() # Neue zufällige Liste wird erzeugt
        product = list_product(current_list) # Das Produkt der Liste wird mithilfe der oben definierten Funktion
herausgefunden
        if product > best_product: # Es wird herausgefunden, ob das aktuelle Produkt das bisher größte ist
            best_product = product # Falls ja, wird das aktuelle Produkt zum besten Produkt
            best_list = current_list # Die zugehörige Liste wird auch gespeichert
            best_runs = runs # Anzahl der Durchläufe, als das bisher beste Produkt erzeugt wurde
        print(current_list, product, best_list, best_product, best_runs, runs) # All diese Daten werden jeden
Durchlauf ausgegeben
```

```
# Ausführen der Hauptfunktion
```

```
if __name__ == '__main__':
    main()
```