

Zahlen anordnen nach Quadrate

Die natürlichen Zahlen von 1 bis 32 sind so im Kreis anzuordnen, dass jeweils die Summe zweier benachbarter Zahlen eine Quadratzahl ist.

Welche Zahl steht der „4“ gegenüber?

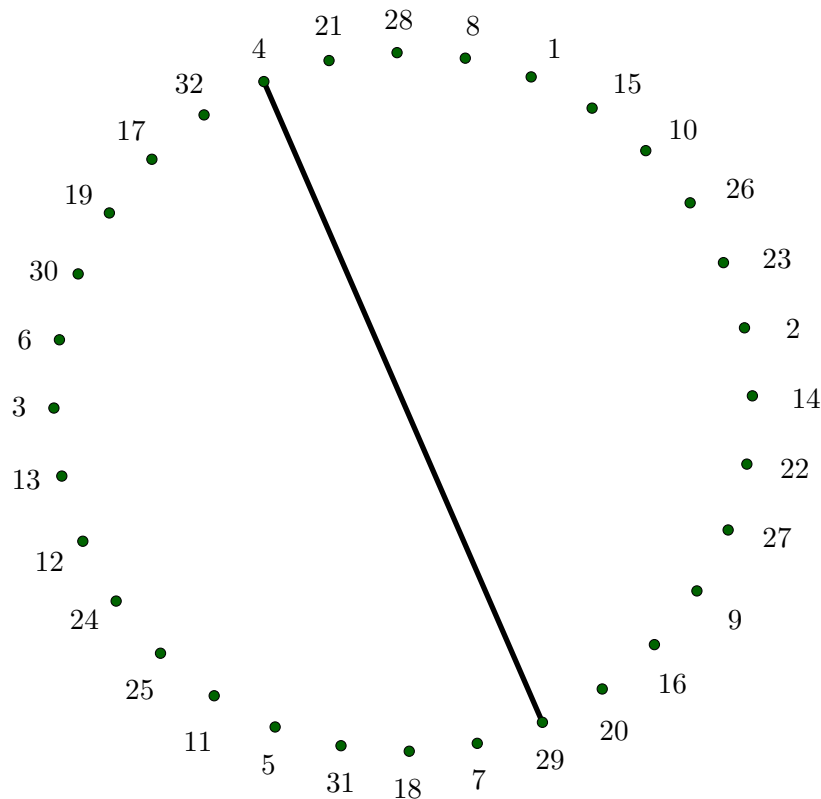
Idee der Aufgabe nach „Die Quadratur der Schachfiguren“, aus Cogito - Bild der Wissenschaft, Heft 11, 2008, von Heinrich Hemme

Lösung

Von jeder Zahl werden die möglichen Nachbarn aufgeschrieben, die als Summe mit der Ausgangszahl eine Quadratzahl wird. So ist

1	3	8	15	24		
2		7	14	23		
3	1	6	13	22		
4		5	12	21	32	
5		4	11	20	31	
6		3	10	19	30	
7		2	9	18	29	
8		1		17	28	
9			7	16	27	
10			6	15	26	
11			5	14	25	
12			4	13	24	
13			3	12	23	
14			2	11	22	
15			1	10	21	
16				9	20	
17				8	19	32
18				7		31
19				6	17	30
20				5	16	29
21				4	15	28
22				3	14	27
23				2	13	26
24				1	12	25
25					11	24
26					10	23
27					9	22
28					8	21
29					7	20
30					6	19
31					5	18
32					4	17

Eindeutige Zuordnungen ergeben dann zehn Dreierkombinationen (9, 16, 20), (7, 18, 31), (11, 24, 25), (10, 26, 23), (9, 27, 22), (8, 28, 21), (7, 29, 20), (6, 30, 19), (5, 31, 18), (4, 32, 17). Da die 9 und die 7 jeweils zweimal in den Dreierkombinationen vorkommen, müssen sie zusammengefasst werden, es entstehen zwei Fünferkombinationen (20, 29, 7, 18, 31) und (20, 16, 9, 27, 22), die wieder zur Neunerkombination (22, 27, 9, 16, 20, 29, 7, 18, 31) ergänzt wird. An das Ende der 31 passt nur die 5, so dass schon zehn Zahlen (22, 27, 9, 16, 20, 29, 7, 18, 31, 5) vergeben sind. Für die verbleibenden sechs Dreierkombinationen und die sieben Zahlen (1, 2, 3, 12, 13, 14, 15) gibt es nur noch wenige Kombinationsmöglichkeiten. Die Lösung ist (14, 2, 23, 26, 10, 15, 1, 8, 28, 21, 4, 32, 17, 19, 30, 6, 3, 13, 12, 24, 25, 11, 5, 31, 18, 7, 29, 20, 16, 9, 27, 22).



Gegenüber der Zahl 4 liegt die Zahl 29.

Ingmar Rubin, Berlin, hat ein wunderbares Programm mit Python geschrieben. Bei zehn Versuchen findet es schon nach einer Sekunde die Lösung mit vier unterschiedlichen Anfangswerten.

```
import random          #Bibliothek für Zufallszahlen wird geladen
k = 10                 #Anzahl der Versuche
n = 32                 #32 Zahlen werden benötigt
m = 8                  #Quadratzahlen bis 72
l2 = list(range(1, n + 1)) #erzeugt eine Liste mit den Zahlen l2 = 1, 2, 3, ..., 32
q = (k * *2 for k in range(2, m)) #erzeugt die Quadratzahlen 4, 9, 16, 25, 36, 49
l3 = list(q)           #erzeugt eine Liste mit den Zahlen l3 = 4, 9, 16, 25, 36, 49
for j in range(k) :    #10 Versuche, um eine Lösung zu finden
    l4 = []             #l4 wird bei jedem neuen Versuch leer gemacht
    while len(l4) < n : #es soll solange, wie möglich probiert werden
        l1 = l2[:]      #liste l2 der 32 Zahlen wird kopiert und heisst l1
        random.shuffle(l1) #mischt die Liste l1 der 32 Zahlen
        z = l1.pop()     #nimmt das letzte Element von l1, setzt es z und löscht es aus l1
        l4 = []          #l4 ist die Lösungsliste, sie soll zu Beginn leer sein
        l4.append(z)      #speichert z in l4
        for i in range(n - 1) : #es sind nur noch 31 Zahlen in l1, mache 31 mal
            for s in l1 :    #s ist die erste Zahl in l1, sie wird betrachtet
                if s + z in l3 : #ist s + z eine Quadratzahl,
                    l4.append(s) #so speichere s in l4 ab, es sind jetzt in l4 = [z, s]
                    l1.remove(s) #lösche s in l1
                    z = s        #setze s = z und überprüfe, ob das nächste s + z eine Quadratzahl ist
                break          #breche ab und nehme die nächste Zahl s in l1
            if l4[0] + l4[n - 1] in l3 : #ist die Summe aus erstem und letztem Element eine Quadratzahl
                print(l4)        #wenn ja, dann drucke die Lösungsliste l4
[31, 18, 7, 29, 20, 16, 9, 27, 22, 14, 2, 23, 26, 10, 15, 1, 8, 28, 21, 4, 32, 17, 19, 30, 6, 3, 13, 12, 24, 25, 11, 5]
[6, 3, 13, 12, 24, 25, 11, 5, 31, 18, 7, 29, 20, 16, 9, 27, 22, 14, 2, 23, 26, 10, 15, 1, 8, 28, 21, 4, 32, 17, 19, 30]
[25, 24, 12, 13, 3, 6, 30, 19, 17, 32, 4, 21, 28, 8, 1, 15, 10, 26, 23, 2, 14, 22, 27, 9, 16, 20, 29, 7, 18, 31, 5, 11]
[1, 15, 10, 26, 23, 2, 14, 22, 27, 9, 16, 20, 29, 7, 18, 31, 5, 11, 25, 24, 12, 13, 3, 6, 30, 19, 17, 32, 4, 21, 28, 8]
```