Task 3: Rekursive Formel für "Josephus Slightly Different"

Zur Ermittlung der Formel für Josephus bei Start=2 und jede 3te Zahl wird gestrichen habe ich anfänglich die Lösung für Start=1 und jede 3te Zahl wird angesehen:

$$n \,\%\, 3 = 0 : J(n) = J\left(\frac{2n}{3}\right) + \left\lfloor \frac{J\left(\frac{2n}{3}\right) - 1}{2} \right\rfloor$$

$$n \,\%\, 3 = 1 : J(n) = \left(\left(J\left(\left\lfloor \frac{2n}{3} \right\rfloor\right) + \left\lfloor \frac{J\left(\left\lfloor \frac{2n}{3} \right\rfloor\right) - 1}{2} \right\rfloor + 3 - 1\right) \,\%\, n\right) + 1$$

$$n \,\%\, 3 = 2 : \ J(n) = J\left(\left\lfloor \frac{2n}{3} \right\rfloor\right) + \left\lfloor \frac{J\left(\left\lfloor \frac{2n}{3} \right\rfloor\right)}{2} \right\rfloor + 1$$

Um nun auf die Neue Formel zu kommen habe ich zuerst die Angabe Formel mit Start= 1 mit den Lösungen von n=1-100 mit den Lösungen bei Start = 2 verglichen.

Dabei fiel mir auf, dass in den meisten Fällen Start = 1 immer nur um 1 größer war als Start = 2:

Start1:

```
1 2 2 1 4 1 4 7 1 4 7 10 13 2 5 8 11 14 17 20 2 5 8 11 14 17 20 23 26 29 1 4 7
10 13 16 19 22 25 28 31 34 37 40 43 46 2 5 8 11 14 17 20 23 26 29 32 35 38 41
44 47 50 53 56 59 62 65 68 1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52
55 58 61 64 67 70 73 76 79 82 85 88 91
```

Start2:

```
1 1 1 4 3 6 3 6 9 3 6 9 12 1 4 7 10 13 16 19 1 4 7 10 13 16 19 22 25 28 31 3 6
9 12 15 18 21 24 27 30 33 36 39 42 45 1 4 7 10 13 16 19 22 25 28 31 34 37 40
43 46 49 52 55 58 61 64 67 70 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51
54 57 60 63 66 69 72 75 78 81 84 87 90
```

Die Lösung war nur dann nicht um eine Zahl verschoben, wenn bei Start:1 eine 1 als letzter überblieb. Was Sinn macht da wir bei Start:2 einen Mann später anfangen und deshalb nicht der erste Mann, sondern der letzte überlebt.

Kleiner Fun Fact am Rande: Die Edge Cases wo bei Start:1 als letzter Mann die Nummer eins überbleibt sind:

```
1, 1, 2, 3, 4, 6, 9, 14, 21, 31, 47, 70, 105, 158, 237, 355, 533, 799,
1199, 1798, 2697, 4046, 6069, 9103, 13655, 20482, 30723, 46085, 69127,
103691, 155536, 233304, 349956, 524934, 787401, 1181102, 1771653, 2657479,
3986219, 5979328, 8968992, 13453488, 20180232, 30270348, 45405522,
68108283, 102162425, 153243637, 229865456, 344798184
```

Schnell mal in oeis.org eingetippt habe ich gefunden:

https://oeis.org/A081614

ich konnte das aber leider nicht interpretieren, aber zumindest gings in den Kommentaren um das Josephus Problem.

Dadurch habe ich folgende Formel konzipiert:

Formulas

for $f(n) == 1$: Josephus_Start_2 = n

for $f(n) \neq 1$: Josephus_Start_2 = $f(n) - 1$

$f(n)$... Formula from ~~prev.~~ Angabe

Da der Edge Case immer nur 1 betrifft: können wir immer davon ausgehen, dass ein Mann weiter links überlebt als bei Start 1

Beim Edge Case Letzter Überlebender = 1, überlebt anstelle des ersten Mannes der letzte Mann

```python
from math import floor


def j(n):
    if n == 1:
        return 1
    elif n%3 == 0:
        temp = j(2*n/3)
        return (int)(temp + floor((temp-1)/2) )
    elif n%3 == 1:
        temp = j(floor(2*n/3))
        return (int)((temp + 3 + floor((temp-1)/2) - 1) % n + 1)
    elif n%3 == 2:
        temp = j(floor(2*n/3))
        return (int)(temp + floor(temp/2) + 1)

def j_start2(n):
    solution = j(n)

    if solution == 1:
        solution = n
    else :
        solution = solution -1
    return solution

solution_rek = ""
solution_brute_force = ""
for x in range(1, 101):
    solution = j_start2(x)
    if(x == 100):
        solution_rek += str(solution)
        solution_brute_force += str(calculate_last_standing_man(2, 3, x))
    else:
        solution_rek += str(solution)+" "
        solution_brute_force += str(calculate_last_standing_man(2, 3, x))+" "

print(solution_rek)
print(solution_brute_force)
solution_rek == solution_brute_force
```

Daraus erhalte ich nun:

Solution_rek:

```
1 1 1 4 3 6 3 6 9 3 6 9 12 1 4 7 10 13 16 19 1 4 7 10 13 16 19 22 25 28 31 3 6
9 12 15 18 21 24 27 30 33 36 39 42 45 1 4 7 10 13 16 19 22 25 28 31 34 37 40
43 46 49 52 55 58 61 64 67 70 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51
54 57 60 63 66 69 72 75 78 81 84 87 90
```

Solution_brute_force:

```
 1 1 1 4 3 6 3 6 9 3 6 9 12 1 4 7 10 13 16 19 1 4 7 10 13 16 19 22 25 28 31 3
6 9 12 15 18 21 24 27 30 33 36 39 42 45 1 4 7 10 13 16 19 22 25 28 31 34 37 40
43 46 49 52 55 58 61 64 67 70 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51
54 57 60 63 66 69 72 75 78 81 84 87 90
```

Also aus der rekursiven und der Brute Force Methode dasselbe Ergebnis ☜(ﾟヮﾟ☜)

Feedback:

Die Lösung ist zwar rekursiv, aber vermutlich nicht in dem Sinne wie es von der Aufgabe gedacht war. Ich habe mich mehrere Stunden damit beschäftigt und auch die Aufzeichnung der Vorlesungen mehrmals angesehen, um vielleicht doch noch auf eine eigene Rekursive Formel zu kommen, dies ist mir leider nicht gelungen, weshalb ich nun die Formel aus der Angabe einfach erweitert habe.

Deadline war zu knapp gewählt.

Außerdem finde ich die In Class Tasks nicht sinnführend, vor allem in Online/Hybridvorlesungen.

Beim ersten In Class Task musste ich zuhause Quarantäne schieben und hatte dadurch einfach Nachteil bei einem In Class Task der von anderen vor Ort in Gruppen durchgeführt wird, und man selbst sowieso schon nicht in der besten gesundheitlichen Form teilnehmen kann.