Number Theory
The joy of recursion / iteration

Hamilton Python Users Group
14 November 2022
Ian Stewart

Number Theory
The joy of Integers !!!

In Python, if integers in then integers out, with these operators:

- + Addition
- - Subtraction
- * Multiplication
- // Integer divide
- % Modulo
- ** Exponent / pow()

Also:

- Reverse/Rearrange the order of digits in an integer
- Sort order of digits in integer
- Sort of integers in a list

Kaprekar Constant

- D. R. Kaprekar
- Recreational mathematician
- 1905 – 1986
- https://en.wikipedia.org/wiki/Kaprekar%27s_routine
- https://en.wikipedia.org/wiki/D._R._Kaprekar

Rules...

Kaprekar's routine is an iterative algorithm that, with each iteration:

- Take a natural number in a given number base.
- Must have at least two distinct digits.
- Create two new numbers by sorting the digits of the number by descending and ascending order.
- Subtract the second from the first to yield the natural number for the next iteration.

```
Natural number in base 10:   164

164 in descending order:     641
164 in ascending order:      146
                            ====
Subtraction:                 495
```

Example...

With three digits loop on 495

```
Natural number in base 10:   495

164 in descending order:     954
164 in ascending order:      459
                            ====
Subtraction:                 495
```

Demo:
python kaprekar_constant.py 164 – 1 iteration
python kaprekar_constant.py 100 – 6 iterations
555 and 495

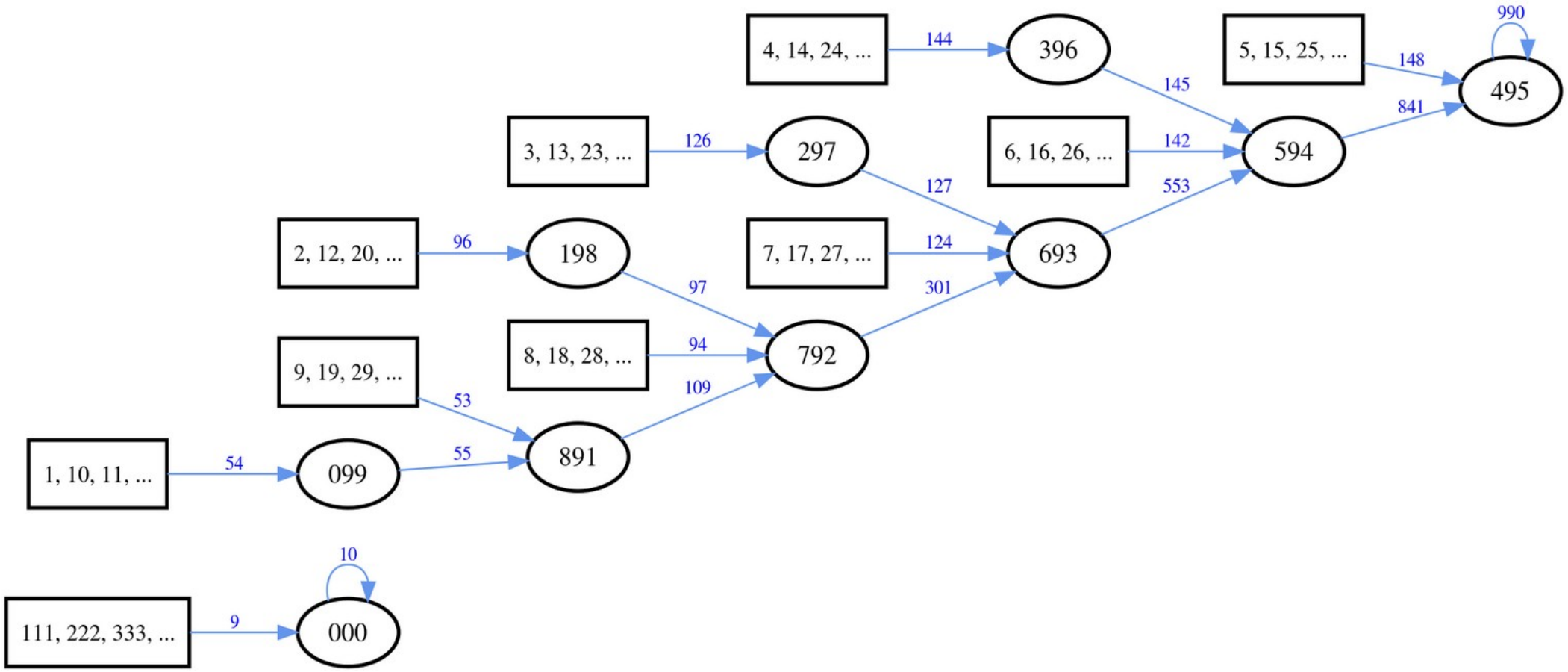
What about a 4 digit number? E.g. 2060

```
Initial integer:              2060

Iteration count:                 1
Rearranged decending:         6200
Rearranged accending:         0026
Sutraction result:            6174

Iteration count:                 2
Rearranged decending:         7641
Rearranged accending:         1467
Sutraction result:            6174
```

Kaprekar process for three digits...



Ian’s Python approach.

- Only one mathematical operation (subtract) per iteration.
- Most of the program concerns digit position manipulaton.
- Keep data as string or string in a list.
- So… Change string data to integer to perform the subtraction.

Ian’s Python approach.

- Convert string to list:

```
>>> int_str = "1234"
>>> int_list  = list(int_str)

>>> int_str
'1234'
>>> int_list
['1', '2', '3', '4']
```

```
>>> int_list = []
>>> int_str = "1234"
>>> int_list[:0] = int_str

>>> int_str
'1234'
>>> int_list
['1', '2', '3', '4']
```

Ian's Python approach.

- Convert string to list using *sorted*:

```
>>> int_str = "2341"
>>>
>>> sorted(int_str)
['1', '2', '3', '4']
```

```
>>> int_str = "2341"
>>>
>>> sorted(int_str, reverse=True)
['4', '3', '2', '1']
```

- Sort list in reverse order:

```
>>> int_list = ["2", "3", "4", "1"]
>>> int_list
['2', '3', '4', '1']

>>> int_list.sort(reverse=True)
>>> int_list
['4', '3', '2', '1']
```

- Convert list back to string:

```
>>> int_descend = "".join(int_list)
>>> int_descend
'4321'
```

Ian’s Python approach.

- Reverse a list:

```
>>> int_list
['4', '3', '2', '1']

>>> int_list.reverse()
>>> int_list
['1', '2', '3', '4']
```

- Reverse a string:

```
>>> int_str
'1234'

>>> int_str[::-1]
'4321'
```

Demo:
python kaprekar_constant.py - 100 to 9999 with their iteration count

Narcissistic Number

Also known as:

- pluperfect digital invariant (PPDI)
- Armstrong number
- plus perfect number

Michael F. Armstrong
1941 – 2020
American mathematician and teacher of computer science

Narcissistic Number

A number that is the sum of its own digits each raised to the power of the number of digits.
E.g. 153_{10} is three digits. So…

$$1^3 + 5^3 + 3^3 = 153$$

$$1 + 125 + 27 = 153$$

There are only 89 narcissistic numbers in base 10, of which the largest is: 115,132,219,018,763,992,565,095,597,973,971,522,401

...with 39 digits.

Narcissistic Number

Get the total digits in a number:

- Iterate integer divide method:

```
>>> y = 370
>>> digit_count = 0

>>> while y > 0:
...     digit_count = digit_count + 1
...     y = y // 10
...
>>> digit_count
3
```

- Length of string method

```
>>> y = 370
>>> digit_count = len(str(y))
>>> digit_count
3
```

Demo:
python narcissistic.py – 26 seconds to get fir less than 1000000

Narcissistic Number

```
#for i in range (0, 1000000): # 1M 2 secs 21 found
for i in range (0, 10000000): # 10M 26 secs 25 found
#for i in range (0, 100000000): # 100M 278 secs 28 found
#for i in range (0, 1000000000): # 1B 3167 secs 32 found
```

Demo:
python narcissistic.py – 26 seconds to get fir less than 1000000

python narcissistic_1.py
Enter number:115132219018763992565095597973971522401

Prime Number

Tkinter program with various methods of finding prime numbers:

- Skips the even numbers. Excessively recursive
- Skip all modulo 6 that do not have a remainder of 1 or 5
- Minimal Recursion
- Use sympy: one at a time. list(sympy.primerange(i, i+1))
- Use sympy: list(sympy.primerange(start_integer, end_integer))

Demo:
python prime_number_locator.py