**Problem # 967.** [Numbers With Same Consecutive Differences](https://leetcode.com/problems/numbers-with-same-consecutive-differences)

<https://leetcode.com/problems/numbers-with-same-consecutive-differences/>

Simple Python 3 solutions using two approaches -- string and integer

I first solved this problem using a string approach. The runtime beats 87.5% However, the integers approach is better and uses less memory as well with runtime beats 97.22%

1. String approach: To get the last digit of the number, I converted the number to a string and then took index = -1.

class Solution:

def numsSameConsecDiff(self, N: int, K: int) -> List[int]:

if N == 1:

return(list(range(10)))

res = [str(x) for x in range(1, 10)]

i = 1 # counter for length if the string

while i < N:

alist = []

for item in res:

num = int(item[-1]) # get the last digit of the number

if K == 0:

alist.append(item + item[-1])

else:

if num - K >= 0:

alist.append(item + str(num - K)) # append the newly calculated digit as string

if num + K < 10:

alist.append(item + str(num + K)) # append the newly calculated digit as string

res = alist

i += 1

return(res)

#----------------------------------------

2. I then thought that working directly with integers will be faster.

So I used the number approach. To get the last digit of a number, I used Modulo operator %.

To shift the number to the left by one digit, I multiplied the number by 10.

class Solution: def numsSameConsecDiff(self, N: int, K: int) -> List[int]: if N == 1: return(list(range(10)))

res = list(range(1, 10))

i = 1

while i < N:

alist = []

for num in res:

if K == 0:

alist.append((num \* 10) + num % 10)

else:

if (num % 10) + K < 10:

alist.append((num \* 10) + (num%10) + K)

if (num % 10) - K >= 0:

alist.append((num \* 10) + (num%10) - K)

res = alist

i += 1

return(res)

===============================================