Nth Digit

Find the n^{th} digit of the infinite integer sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...

Note:
n is positive and will fit within the range of a 32-bit signed integer (n 31).

Example 1:

```
Input:
3

Output:
3
```

Example 2:

```
Input:
11

Output:
0

Explanation:
The 11th digit of the sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ... is a 0, which
 is part of the number 10.
```

Subscribe to see which companies asked this question

Solution 1

Straight forward way to solve the problem in 3 steps:

1. find the length of the number where the nth digit is from
2. find the actual number where the nth digit is from
3. find the nth digit and return

```
public int findNthDigit(int n) {
 int len = 1;
 long count = 9;
 int start = 1;

 while (n > len * count) {
  n -= len * count;
  len += 1;
  count *= 10;
  start *= 10;
 }

 start += (n - 1) / len;
 String s = Integer.toString(start);
 return Character.getNumericValue(s.charAt((n - 1) % len));
}
```

written by lzb700m original link here

Solution 2

Check the same-length ranges 1-9, 10-99, 100-999, 1000-9999, etc.

Python:

```python
def findNthDigit(self, n):
    n -= 1
    for digits in range(1, 11):
        first = 10**(digits - 1)
        if n < 9 * first * digits:
            return int(str(first + n/digits)[n%digits])
        n -= 9 * first * digits
```

Java:

```java
public int findNthDigit(int n) {
    n -= 1;
    int digits = 1, first = 1;
    while (n / 9 / first / digits >= 1) {
        n -= 9 * first * digits;
        digits++;
        first *= 10;
    }
    return (first + n/digits + "").charAt(n%digits) - '0';
}
```

Using divisions instead of multiplications to prevent overflow.

written by StefanPochmann original link here

Solution 3

```
We are tightening the range of our target gradually.
1. Is this number that has nth digit between 100~999, 1000~9999 or some other rang
e?
2. What is this number?
3. Within thin number, which digit is the nth digit?

number of the digits at each "level"
1-9: 9 digits
10-99 : 90 * 2 = 180 digits
100-999 : 900 * 3 = 2700 digits
1000-9999 : 9000 * 4 = 36000 digits
....
*/
public class Solution {
    public int findNthDigit(int n) {
        if(n < 1) return 0;
        if(n < 10) return n;
        int counter = 1;  //stores the level number
        int base = 0;      //stores the biggest number from previous level
        while(n > (9 * Math.pow(10,counter -1) * (counter))){
            base += 9 * Math.pow(10,counter -1);
            n -= (9 * Math.pow(10,counter -1) * (counter));
            counter++;
        }
        //target is the actual number that has nth digit
        int target = base + ((n + counter - 1) / counter);  //to get the ceiling
of n / counter
        int offset = n % counter;
        offset = (offset == 0) ? 0:counter - offset;
        for(int i = 0; i < offset; i++){
            target = target / 10;
        }
        return target % 10;
    }
}
```

written by DonaldTrump original link here