Find n'th number in a number system with only 3 and 4

Given a number system with only 3 and 4. Find the nth number in the number system. First few numbers in the number system are: 3, 4, 33, 34, 43, 44, 333, 334, 343, 344, 433, 434, 443, 444, 3333, 3334, 3343, 3344, 3433, 3434, 3443, 3444, …

Source: Zoho Interview

We strongly recommend to minimize the browser and try this yourself first.

We can generate all numbers with i digits using the numbers with (i-1) digits. The idea is to first add a '3' as prefix in all numbers with (i-1) digit, then add a '4'. For example, the numbers with 2 digits are 33, 34, 43 and 44. The numbers with 3 digits are 333, 334, 343, 344, 433, 434, 443 and 444 which can be generated by first adding a 3 as prefix, then 4.

Following are detailed steps.

```
1) Create an array 'arr[]' of strings size n+1.
2) Initialize arr[0] as empty string. (Number with 0 digits)
3) Do following while array size is smaller than or equal to n
.....a) Generate numbers by adding a 3 as prefix to the numbers generated
        in previous iteration.  Add these numbers to arr[]
.....a) Generate numbers by adding a 4 as prefix to the numbers generated
        in previous iteration. Add these numbers to arr[]
```

Thanks to kaushik Lele for suggesting this idea in a comment here. Following is C++ implementation for the same.

```
// C++ program to find n'th number in a number system wi
#include <iostream>
using namespace std;

// Function to find n'th number in a number system with
void find(int n)
{
    // An array of strings to store first n numbers. arr
    string arr[n+1];
    arr[0] = ""; // arr[0] stores the empty string (Stri

    // size indicates number of current elements in arr[
    // number of elements added to arr[] in previous ite
```

```
    int size = 1, m = 1;

    // Every iteration of following loop generates and a
    // arr[] using the m numbers generated in previous i
    while (size <= n)
    {
        // Consider all numbers added in previous iterat
        // "3" to them and add new numbers to arr[]
        for (int i=0; i<m && (size+i)<=n; i++)
            arr[size + i] = "3" +  arr[size - m + i];

        // Add prefix "4" to numbers of previous iterati
        // numbers to arr[]
        for (int i=0; i<m && (size + m + i)<=n; i++)
            arr[size + m + i] = "4" +  arr[size - m + i]

        // Update no. of elements added in previous iter
        m = m<<1; // Or m = m*2;

        // Update size
        size = size + m;
    }
    cout << arr[n] << endl;
}

// Driver program to test above functions
int main()
{
    for (int i = 1; i < 16; i++)
        find(i);
    return 0;
}
```

Output:

```
3
4
33
34
43
44
333
334
343
344
433
```

```
434
443
444
3333
```