3304. Find the K-th Character in String Game I

Solved

Easy | Topics | Companies | Hint

Alice and Bob are playing a game. Initially, Alice has a string `word = "a"`.

You are given a **positive** integer `k`.

Now Bob will ask Alice to perform the following operation **forever**:

- Generate a new string by **changing** each character in `word` to its **next** character in the English alphabet, and **append** it to the *original* `word`.

For example, performing the operation on `"c"` generates `"cd"` and performing the operation on `"zb"` generates `"zbac"`.

Return the value of the k^{th} character in `word`, after enough operations have been done for `word` to have **at least** `k` characters.

Note that the character `'z'` can be changed to `'a'` in the operation.

Example 1:

Input: k = 5

Output: "b"

Explanation:

Initially, `word = "a"`. We need to do the operation three times:

- Generated string is `"b"`, `word` becomes `"ab"`.
- Generated string is `"bc"`, `word` becomes `"abbc"`.
- Generated string is `"bccd"`, `word` becomes `"abbcbccd"`.

Python3 • Auto

```python
class Solution:
    def kthCharacter(self, k: int) -> str:
        def dfs(k):
            if k == 1:
                return 'a'

            l = 1
            while l * 2 < k:
                l *= 2

            if k <= l:
                return dfs(k)
            else:
                prev = dfs(k - l)
                return chr(ord(prev) - ord('a') + 1 % 26 + ord('a'))

        return dfs(k)
```

Ln 15, Col 69 | Saved

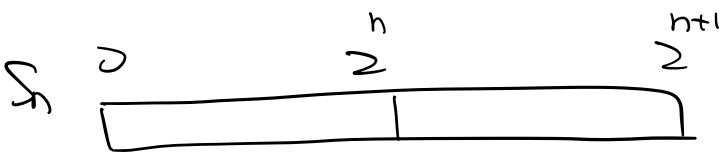
Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

k =



if $0 \le k < 2^n$:

reduce directly

else:

reduce with $(k - 1/2 \text{ length})$

then shift

Method 2: Bitwise & Mathematical

```
def kthCharacter(k):
    ans = 0                                  # number of shifts
    while (k != 1):
        t = k.bit_length() - 1               # find highest power of 2 <= k.
                                             # 2^t <= k, find t
        if (1 << t) == k:                    # edge correction
```

```
        t -= 1
    k -= 1 << t
    ans += 1
return chr(ord('a') + ans % 26)
```

if k is exactly power of 2, it means it's the first index of the shift part, so step back one level.

↳ move to previous half