yuhan210 ★ 12 Last Edit: September 14, 2018 9:45 AM 1.1K VIEWS

12

Segment tree implementation.
Space complexity: O(2n)
Time complexity: get O(2logn); check: O(1); release: O(logn).

```
class PhoneDirectory(object):

    def __init__(self, max_number):
        self.tree = [True] * 2 * max_number
        self.max_number = max_number


    def get(self):
        if self.tree[1] == False: return -1
        i = 1
        while i < len(self.tree)/2:
            if 2 * i < len(self.tree) and self.tree[2 * i]:
                i = 2 * i
            if 2 * i + 1 < len(self.tree) and self.tree[2 * i + 1]:
                i = 2 * i + 1

        ret = i - self.max_number

        # update the tree
        self.tree[i] = False
        i /= 2
        while i > 0:
            self.tree[i] = self.tree[2 * i] or self.tree[2 * i + 1]
            i /= 2

        return ret

    def check(self, number):
        return number >= 0 and number < self.max_number and self.tree[number + self.max_number]

    def release(self, number):
        i = self.max_number + number
        while i > 0:
            self.tree[i] = True
            i /= 2
```