Question - You are a product manager and currently leading a team to develop a new product. Unfortunately, the latest version of your product fails the quality check. Since each version is developed based on the previous version, all the versions after a bad version are also bad.

Suppose you have n versions [1, 2, ..., n] and you want to find out the first bad one, which causes all the following ones to be bad.

You are given an API bool isBadVersion(version) which will return whether version is bad. Implement a function to find the first bad version. You should minimize the number of calls to the API.

Solution -

class Solution:

def firstBadVersion(self, n)

return badversion(n,1,n)

def badversion(n,left,right):

while left <= right:

middle = (left + right) // 2

firstbadversion = isBadVersion(middle)

onelag = isBadVersion(middle-1)

if firstbadversion and not onelag:

return middle

elif firstbadversion:

right = middle-1

else:

left = middle + 1

return -1