

1. Compute and return the square root of x , where x is guaranteed to be a non-negative integer. And since the return type is an integer, the decimal digits are truncated and only the integer part of the result is returned. Also, talk about the time complexity of your code.

Test Cases:

Input: 4

Output: 2

Input: 8

Output: 2

Explanation: The square root of 8 is 2.8284...., the decimal part is truncated and 2 is returned.

Hint: Try to use a modified binary search approach for an optimized solution

2. 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 version and you want to find out the first bad one, which causes all the following ones to be bad. Also, talk about the time complexity of your code.

Test Cases:

Input: [0,0,0,1,1,1,1,1]

Output: 3

Explanation: 0 indicates a good version and 1 indicates a bad version. So, the index of the first 1 is at 3. Thus, the output is 3

3. Given a positive integer num , write a program that returns True if num is a perfect square else return False. Do not use built-in functions like `sqrt`. Also, talk about the time complexity of your code.

Test Cases:

Input: 16

Output: True

Input: 14

Output: False