Data Structure and Algorithms

Rafiq Islam

Table of contents

## Binary Search

1. Leetcode 69: Sqrt(x)
* Given a non-negative integer, $x$, return the square root of $x$ rounded down to the nearest integer. The returned integer should be non-negative as well.
* You may not use any built-in exponent function. For example, x\*\*0.5 in python.
* Example:
* Input: x=4
 Output: 2

 Input: x=8
 Output: 2
* Explanation: Square root of 4 is 2 and square root of 8 is 2.8284. But we need to round down to any fraction. Therefore, the square root of 8 is also 2.
* **Solution:**
* The square root of any number $x\geq 0$ is less than or equal to $x$. The brute force solution to this would be $O\left(\sqrt{n}\right)$. Because, say $x=8$, then
* for $i=1$ to 8:
* 
* In contrast, if we explore binary search then the time complexity reduces to $O\left(logn\right)$. Say the square root is $s$ which is the middle value in the range of 1 to $x$. Then if $s^{2}>x$, we search for the root in the left half. Otherwise, if $s^{2}<x$ then we search the right side. However, when $s^{2}<x$, then $s$ is a possible candidate for the square root.
* *Algorithm:*
	1. set left value $l=0$, right value $r=x$
	2. Compute the middle value $m=l+\left(r−l\right)/2$
	3. If $m^{2}>x$ then search the left side: set $r=m−1$
	4. If $m^{2}<x$ then search the right side: set $l=m+1$
* def square\_root(x):
 l, r = 0, x
 sq = 0
 while l<=r:
 m = l + (r-l)//2
 if m\*\*2 > x:
 r= m-1
 elif m\*\*2 < x:
 l = m+1
 sq = m
 else:
 return m
 return sq

 print(square\_root(6))
* 2
1. item
2. item