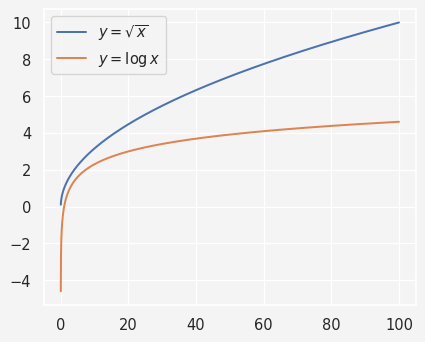
Data Structure and Algorithms

Rafiq Islam

Table of contents

## Binary Search

1. Leetcode 69: Sqrt(x)

* Given a non-negative integer, , return the square root of 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 is less than or equal to . The brute force solution to this would be . Because, say , then
* for to 8:
* 
* In contrast, if we explore binary search then the time complexity reduces to . Say the square root is which is the middle value in the range of 1 to . Then if , we search for the root in the left half. Otherwise, if then we search the right side. However, when , then is a possible candidate for the square root.
* *Algorithm:*
  1. set left value , right value
  2. Compute the middle value
  3. If then search the left side: set
  4. If then search the right side: set
* 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