
Valid Perfect Square

Given a positive integer *num*, write a function which returns True if *num* is a perfect square else False.

Note: Do not use any built-in library function such as `sqrt`.

Example 1:

Input: 16
Returns: True

Example 2:

Input: 14
Returns: False

Credits:

Special thanks to [@elmirap](#) for adding this problem and creating all test cases.

Solution 1

```
public boolean isPerfectSquare(int num) {  
    int i = 1;  
    while (num > 0) {  
        num -= i;  
        i += 2;  
    }  
    return num == 0;  
}
```

written by [fhqplzj](#) original link [here](#)

Solution 2

1. a square number is $1+3+5+7+\dots$ Time Complexity $O(\sqrt{N})$ (Credit to lizhibupt, thanks for correcting this).
2. binary search. Time Complexity $O(\log N)$
3. Newton Method. See [this wiki page](#). Time Complexity is close to constant, given a positive integer.

```
public boolean isPerfectSquare(int num) {
    if (num < 1) return false;
    for (int i = 1; num > 0; i += 2)
        num -= i;
    return num == 0;
}

public boolean isPerfectSquare(int num) {
    if (num < 1) return false;
    long left = 1, right = num; // long type to avoid 2147483647 case

    while (left <= right) {
        long mid = left + (right - left) / 2;
        long t = mid * mid;
        if (t > num) {
            right = mid - 1;
        } else if (t < num) {
            left = mid + 1;
        } else {
            return true;
        }
    }

    return false;
}

boolean isPerfectSquare(int num) {
    if (num < 1) return false;
    long t = num / 2;
    while (t * t > num) {
        t = (t + num / t) / 2;
    }
    return t * t == num;
}
```

written by [coolguy](#) original link [here](#)

Solution 3

Just slightly modified my [sqrt solutions](#). You can find some explanation there.

(Note I renamed the parameter to x because that's the name in the sqrt problem and I like it better.)

Java, C++, C, C#

```
long r = x;
while (r*r > x)
    r = (r + x/r) / 2;
return r*r == x;
```

Python

```
r = x
while r*r > x:
    r = (r + x/r) / 2
return r*r == x
```

Ruby

```
r = x
r = (r + x/r) / 2 while r*r > x
r*r == x
```

JavaScript

```
r = x;
while (r*r > x)
    r = ((r + x/r) / 2) | 0;
return r*r == x;
```

written by [StefanPochmann](#) original link [here](#)

From [LeetCoder](#).