PDP ONLINE

DSA - Algorithms

Number 2

Course Planning

Algorithms	Data Structures	Algorithmic Approaches	Interview Practices
1.Introduction	1.Asymptotic Analysis	1.Search Algorithms	1.In-place Reversal
2.Number 1	2.Dynamic Array	2.Sort Algorithms	2.Two Heaps
3.Number 2	3.LinkedList	3.Dac Algorithms	3.Subsets
4.String 1	4.Stack	4.Recursion	4.Modified BS
5.String 2	5.Queue	5.Sliding Window	5.Bitwise XOR
6.Array 1	6.Tree	6.Two Pointers	6.Top 'K' Elements
7.Array 2	7.Heap	7.Fast & Slow	7.K-way Merge
8.Matrix	8.Trie	8.Cyclic Sort	8.Knapsack Problem
9.DP 1	9.Graph	9.Breadth First Search	9.Topological Sort
10.DP 2	10.Undirected Graph	10.Depth First Search	10.Mock Interview

https://kurbanovxurshid.medium.com/data-structures-and-algorithms-course-plan-1ab912ec1e17

Asked by Microsoft




Explanation

367. Valid Perfect Square

Easy 1267 194 Add to List Share

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

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

Example 1:

```
Input: num = 16
Output: true
```

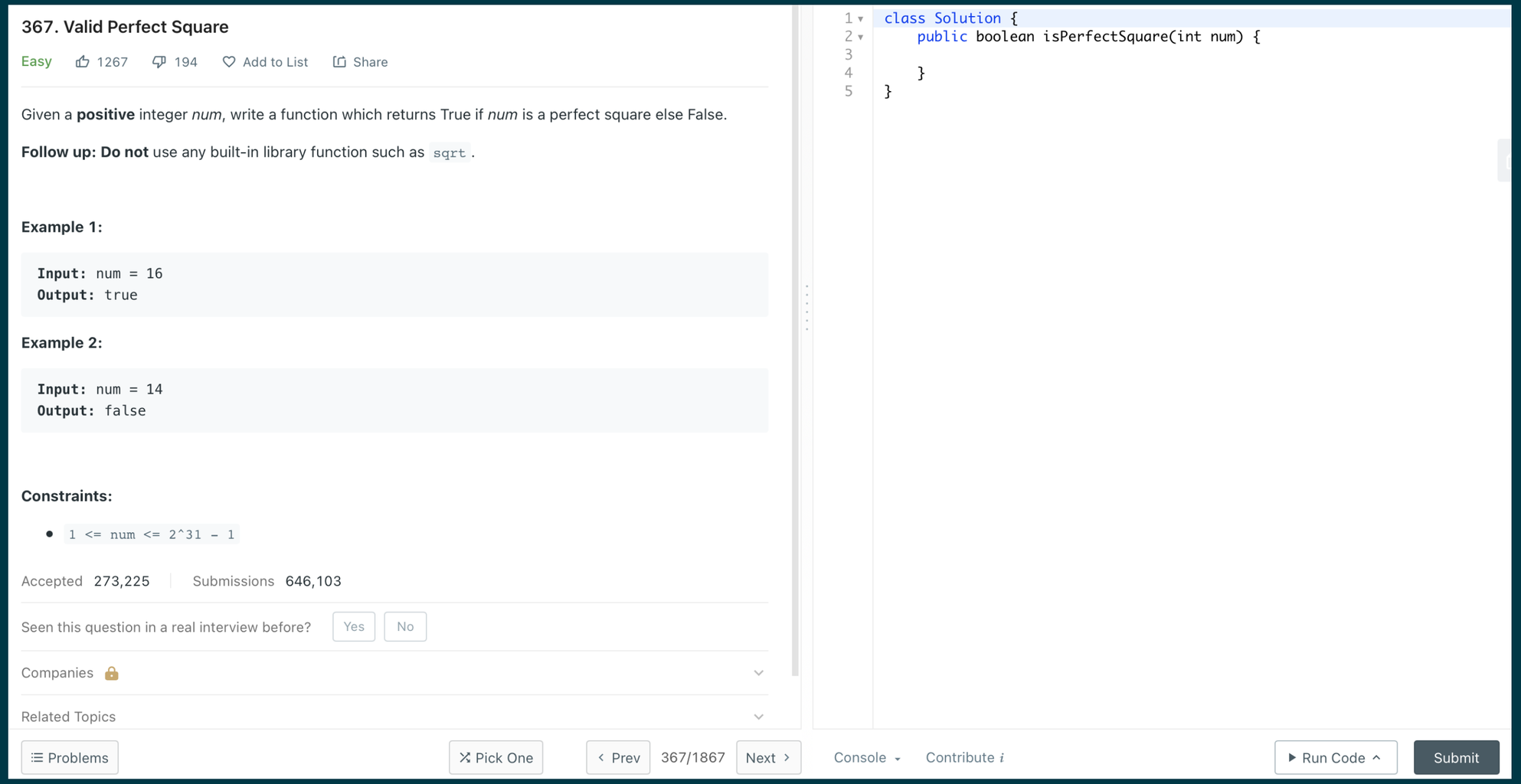
Example 2:

```
Input: num = 14
Output: false
```

Constraints:

- `1 <= num <= 2^31 - 1`

Valid Perfect Square



367. Valid Perfect Square

Easy 1267 194 Add to List Share

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

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

Example 1:

```
Input: num = 16
Output: true
```

Example 2:

```
Input: num = 14
Output: false
```

Constraints:

- `1 <= num <= 2^31 - 1`

Accepted 273,225 Submissions 646,103

Seen this question in a real interview before? Yes No

Companies

Related Topics

```
class Solution {
    public boolean isPerfectSquare(int num) {

    }
}
```

Problems Pick One Prev 367/1867 Next Console Contribute i Run Code Submit

https://leetcode.com/problems/valid-perfect-square/

First Theory

$$1+3+5+7+ \ldots + 2n-1 = n^2$$

$$1+3+5+7+9 = 25 = 5^2$$

First Solution

Success Details ›

Runtime: 0 ms, faster than 100.00% of Java online submissions for Valid Perfect Square.

Memory Usage: 35.7 MB, less than 43.08% of Java online submissions for Valid Perfect Square.

Next challenges:

Sqrt(x) | Sum of Square Numbers

Show off your acceptance:

```java
class Solution {
    public boolean isPerfectSquare(int num) {

        int i = 1;

        while(num > 0){
            num = num - i;
            i = i + 2;
        }

        return num == 0;
    }
}
```

Second Theory

Newton`s Method

n = x^2

x^2 - n = 0

x = (x + n/x) / 2

$$x_0 \approx \sqrt{S},$$

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{S}{x_n}\right),$$

$$\sqrt{S} = \lim_{n \to \infty} x_n.$$

Second Solution

Success Details ›

Runtime: 0 ms, faster than 100.00% of Java online submissions for Valid Perfect Square.

Memory Usage: 35.6 MB, less than 59.54% of Java online submissions for Valid Perfect Square.

Next challenges:

Sqrt(x) Sum of Square Numbers

Show off your acceptance:

```java
class Solution {
    public boolean isPerfectSquare(int num) {

        long x = num;

        while(x * x > num){
            x = (x + num/x) / 2;
        }
        return x*x == num;
    }
}
```

Task 1 – Perfect Number

507. Perfect Number

Easy 394 741 Add to List Share

A **perfect number** is a **positive integer** that is equal to the sum of its **positive divisors**, excluding the number itself. A **divisor** of an integer `x` is an integer that can divide `x` evenly.

Given an integer `n`, return `true` *if* `n` *is a perfect number, otherwise return* `false`.

Example 1:

```
Input: num = 28
Output: true
Explanation: 28 = 1 + 2 + 4 + 7 + 14
1, 2, 4, 7, and 14 are all divisors of 28.
```

Example 2:

```
Input: num = 6
Output: true
```

Example 3:

```
Input: num = 496
Output: true
```

Example 4:

https://leetcode.com/problems/perfect-number/

Task 2 – Ugly Number

263. Ugly Number

Easy 782 798 Add to List Share

An **ugly number** is a positive integer whose prime factors are limited to `2`, `3`, and `5`.

Given an integer `n`, return `true` *if* `n` *is an* ***ugly number***.

Example 1:

```
Input: n = 6
Output: true
Explanation: 6 = 2 × 3
```

Example 2:

```
Input: n = 8
Output: true
Explanation: 8 = 2 × 2 × 2
```

Example 3:

```
Input: n = 14
Output: false
Explanation: 14 is not ugly since it includes the prime factor 7.
```

Example 4:

```
Input: n = 1
Output: true
```

https://leetcode.com/problems/ugly-number/

Task 3 – Add Digits

258. Add Digits

Easy 1190 1309 Add to List Share

Given an integer `num`, repeatedly add all its digits until the result has only one digit, and return it.

Example 1:

```
Input: num = 38
Output: 2
Explanation: The process is
38 --> 3 + 8 --> 11
11 --> 1 + 1 --> 2
Since 2 has only one digit, return it.
```

Example 2:

```
Input: num = 0
Output: 0
```

Constraints:

- $0 <= num <= 2^{31} - 1$

https://leetcode.com/problems/add-digits/