

Sum of Square Numbers

Given a non-negative integer c , your task is to decide whether there're two integers a and b such that $a^2 + b^2 = c$.

Example 1:

Input: 5

Output: True

Explanation: $1 * 1 + 2 * 2 = 5$

Example 2:

Input: 3

Output: False

Solution 1

```
public class Solution {  
    public boolean judgeSquareSum(int c) {  
        if (c < 0) {  
            return false;  
        }  
        int left = 0, right = (int)Math.sqrt(c);  
        while (left <= right) {  
            int cur = left * left + right * right;  
            if (cur < c) {  
                left++;  
            } else if (cur > c) {  
                right--;  
            } else {  
                return true;  
            }  
        }  
        return false;  
    }  
}
```

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

Solution 2

```
def judgeSquareSum(self, c):  
    def is_square(N):  
        return int(N**.5)**2 == N  
  
    return any(is_square(c - a*a)  
               for a in xrange(int(c**.5) + 1))
```

Without loss of generality, let's consider only $a, b \geq 0$. This is because if say, $a < 0$, then we may also find a solution using $\text{abs}(a)$.

Now, $a*a = c - b*b \leq c$, because $b*b \geq 0$, and $0 \leq a \leq \text{sqrt}(c)$ is a necessary condition for a solution.

Let's try each $0 \leq a \leq \text{sqrt}(c)$. For each choice of a , we must have $b*b = c - a*a$. There will be a solution if and only if the right-hand-side is a perfect square.

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

Solution 3

```
public static boolean judgeSquareSum(int c) {  
    for (int i=0;i<=Math.sqrt(c);i++)  
        if (Math.floor(Math.sqrt(c-i*i)) == Math.sqrt(c-i*i)) return true;  
    return false;  
}
```

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

From [LeetCoder](#).