## Algorithm

Check whether a given number is Fibonacci number

## Input

‘n’ a number to check whether is Fibonacci

## Output

A Boolean that indicates the given number is Fibonacci or not

## Method

is\_perfect\_square(n) {

sr = sqrt(n);

sr = floor(sr);

return n == pow(sr \*\* 2);

}

is\_fibonacci(n) {

p = 5 \* pow(n, 2);

return is\_perfect\_square(p + 4) or is\_perfect\_square(p - 4);

}

## Analysis

The number of operations does not depend on the input and is always constant.

So, time complexity is Θ (1)