4.

Algorithm iterativeFibonacci (n):

Input: a natural number n

Output: Fibonacci number at n^{th} position

```
if (n <= 1) return n
a ← 0
fibo ← 1
for i ← 2 to n do
    b ← fibo
    fibo ← a + fibo
    a ← b

return fibo
```

4.

Algorithm iterativeFibonacci (n):

Input : a natural number n

Output : Fibonacci number at n^{th} position

```
if (n <= 1) return n           2
a ← 0                          1
fibo ← 1                       1
for i ← 2 to n do              n-2
    b ← fibo                   1
    fibo ← a + fibo            2
    a ← b                      1
return fibo                    1
```

Running time

$$T(n) = \begin{cases} 2, & n \leq 1 \\ 4(n-2)+3, & n > 1 \end{cases}$$

i.e. $T(n) = 4n - 5, n > 1$

$\therefore$ $T(n)$ is $O(n)$. Here, the recursive version of fibo calculation had time complexity of $O(2^n)$ whereas the iterative version has $O(n)$. It can be made $O(\log n)$ by using matrix multiplication, as well.

Algorithm iterativeFibonacci (n):

Input : a natural number n
Output : Fibonacci number at n^{th} position
if (n <= 1) return n
$a \leftarrow 0$
$fibo \leftarrow 1$
for $i \leftarrow 2$ to n do
  $b \leftarrow fibo$
  $fibo \leftarrow fibo + a$
  $a \leftarrow b$

return fibo

Proof of correctness

We verify this algorithm using the principle of mathematical induction with the loop invariant 'fibo'.

Base case:

After the iteration $i = 1$ completes,
$fibo = 1$

Induction Step:

We assume iterativeFibonacci (K) holds true, & show iterativeFibonacci (K+1) also holds.
At the end of $i = k$ pass,

$$fibo_K = fibo_{K-1} + a_{K-1} \Rightarrow a_{K-1} = fibo_K - fibo_{K-1} = fibo_{K-2}$$

then at $i = K+1$, we have,

$$fibo_{K+1} = fibo_K + a_K$$
$$= fibo_K + fibo_{K-1}$$

which is correct as per the defn of fibonacci sequence.
$\therefore$ The algorithm correctly computes the fibo sequence for $1 \le K \le n$.

Proved.