

2. **Input** $(n) \in \mathbb{N}$

Function *fibonacci* (n) {

if $n = 0$ **return** 1;

else

$a = 1, b = 1, i = 2$

while $i \leq n$ **do**

$c = a + b, a = b, b = c, i++;$

return b ;

}

Chứng minh $\text{fibonacci}(n) = F(n)$

$$F(0) = 0$$

$$F(1) = 1$$

$$F(n) = F(n-1) + F(n-2)$$

$$\text{TH: } n = 0: \Rightarrow \begin{cases} F(n) = 0 \\ \text{fibonacci}(n) = 0 \end{cases} \Rightarrow \text{thỏa TH } n = 0$$

$$\text{TH: } n = 1: \Rightarrow \begin{cases} F(n) = 1 \\ \text{fibonacci}(n) = 1 \text{ (chưa thực hiện vòng lặp)} \end{cases} \Rightarrow \text{thỏa TH } n = 1$$

$$\text{TH: } n = 2: \Rightarrow \begin{cases} F(2) = F(0) + F(1) = 1 \\ \text{fibonacci}(2) = 2 \end{cases}$$

❖ Giả sử thuật toán đúng với các TH $k \leq n$

$$\Rightarrow \text{fibonacci}(k) = F(k)$$

- Vòng lặp *while* thực hiện $(n - i + 1)$ lần = k lần

- Khi vòng lặp tới lần $k - 1$ $i \leq k$ thì

$$a = F(k-1)$$

$$b = F(k)$$

- Lặp đến lần thứ k : $i \leq k + 1$ thì

$$b = F(k) + F(k-1) = F(k+1)$$

\Rightarrow ĐPCM

3. **Iterative maximum**

Input $(n \geq 1)$

Function *maximum* (A, n) {

$m = A[1], i = 2$

while $(i \leq n)$ **do**

if $A[i] > m$ **then** $m = A[i]$

$i = i + 1;$

return m

}

Chứng minh: $\text{maximum}(A, n) = \max(A[1 \dots n])$

$$\text{TH: } n = 1: \max(A[1 \cdots n]) = A(1)$$

$$\text{maximum}(A, 1) = A(1) = m$$

$$\text{TH: } n = 2: \max(A[1 \cdots n]) = \max(A(1), A(2)) = \begin{bmatrix} A(1) \\ A(2) \end{bmatrix}$$

$$\text{maximum}(A, 2):$$

$$m = A(1); i = 2 \leq n = 2$$

$$A[i] > m \Rightarrow m = A[i] = A(2) \text{ nếu } A(2) > A(1)$$

$$i = i + 1 = 3$$

$$\Rightarrow \text{maximum}(A, 2) = \begin{bmatrix} A(1) \\ A(2) \end{bmatrix} = \max(A[1 \cdots 2])$$

$$\text{TH: } n = 3: \max(A[1 \cdots n]) = \max(A(1), A(2), A(3)) = \begin{bmatrix} A(1) \\ A(2) \\ A(3) \end{bmatrix}$$

$$\text{maximum}(A, 3):$$

$$m = A[1]; i = 2$$

$$i \leq n = 3$$

$$\text{Nếu } A[i] > m \Rightarrow m = A[i] = A(2)$$

$$\text{Nếu } A(2) > A(1) \Rightarrow m = \max(A(1), A(2))$$

$$i = i + 1 = 3 \leq n = 3$$

$$\text{Nếu } A[i] > m$$

$$\Rightarrow m = A[i] = A(3) \text{ nếu } A(3) > \max(A(1), A(2))$$

$$\Rightarrow m = \max(A(1), A(2), A(3))$$

$$i = i + 1 = 4 > n \Rightarrow \text{dừng}$$

$$\Rightarrow \text{maximum}(A, 3) = \begin{bmatrix} A(1) \\ \cdots \\ A(3) \end{bmatrix} = \max(A[1 \cdots 3])$$

- ❖ Giả sử thuật toán đúng với các trường hợp $k \leq n$
 $\Rightarrow \text{maximum}(A, k) = \max(A[1 \cdots k])$

Xét TH: $n = k + 1$

- Vòng lặp while thực hiện $k - i + 1 = k - 2 + 1 = (k - 1)$ lần
- Tại lần thứ $(k - 2) \rightarrow$ đang xét vị trí (k)
 $A[i] > m \Rightarrow m = A[k]$ nếu $A[k] > \max(A[1 \dots k - 1])$
- Tại lần thứ $(k - 1)$ vòng lặp sẽ thực hiện việc so sánh
 $A[i] > m \Rightarrow m = A[k + 1]$ nếu $A[k + 1] > \max(A[1 \dots k])$
- Vậy kết quả sẽ có được là $\begin{bmatrix} A[k + 1] \\ \max(A[1 \dots k]) \end{bmatrix} = \max(A[1 \dots k + 1])$

$\Rightarrow \text{maximum}(A, k + 1) = \max(A[1 \dots k + 1])$

\Rightarrow ĐPCM

