

Insertion Sort

ejaramillos

September 2018

1 Fibonacci Código en Java

Utilizando un tipo de dato entero el programa funciona hasta el dígito 47 antes de hacer overflow

Utilizando un tipo de dato long el programa funciona hasta el dígito 92 antes de hacer overflow

2 Fibonacci Código en C

Utilizando un tipo de dato entero el programa funciona hasta el dígito 47 antes de hacer overflow

Utilizando un tipo de dato long el programa funciona hasta el dígito 93 antes de hacer overflow

Utilizando un tipo de dato long long el programa funciona hasta el dígito 93 antes de hacer overflow

3 Cormen 2.1-2

Rewrite the INSERTION-SORT procedure to sort into nonincreasing instead of nondecreasing order.

Algorithm 1 Insertion Sort nonincreasing

```
1: procedure INSERTION SORT( $A$ )
2:   for  $j \leftarrow 2$  to  $A.length$  do
3:      $key = A[j]$ 
4:      $i = j - 1$ 
5:     while  $r > 0$  and  $A[j] < key$  do
6:        $A[i + 1] \leftarrow A[i]$ 
7:        $i = i - 1$ 
8:      $A[i + 1] = key$ 
```

4 Cormen 2.1-3

Write pseudocode for linear search, which scans through the sequence, looking for v . Using a loop invariant, prove that your algorithm is correct. Make sure that your loop invariant fulfills the three necessary properties.

Invariante: En cada iteración del for, el subarreglo que va desde el inicio hasta $A[i-1]$ no contiene el valor del elemento v .

Inicialización: Inicialmente el subarreglo que va desde el inicio hasta $A[i-1]$ es un arreglo vacío.

Mantenimiento: En cada paso i del ciclo Se sabe que desde 0 hasta $A[i]$ el elemento v no está. Si $A[i]$ es igual al valor v , entonces se retorna el valor de i que es el índice del arreglo en el que aparece el valor v por lo tanto la invariante se preserva.

Terminación: Si desde 0 hasta $A[i]$ no se encontró el valor del elemento v al algoritmo retorna NIL, en este caso se preserva la invariante puesto que el subarreglo que va desde el inicio hasta $A[i-1]$ no contiene el valor del elemento v .

Algorithm 2 Linear Search

```
1: procedure LINEAR SEARCH( $A, V$ )
2:   for  $i \leftarrow 0$  to  $A.length$  do
3:     if  $A[i] == V$  then
4:       return  $i$ 
5:   return  $NIL$ 
```

5 Cormen 2.1-4

State the problem formally and write pseudocode for adding the two integers.

Algorithm 3 BinarySum

```
1: procedure BINARY SUM( $A, B$ )
2:    $C = \text{Array}[A.length + 1]$ 
3:    $aux = 0$ 
4:   for  $i \leftarrow 1$  to  $A.length$  do
5:      $C[i] = (A[i] + B[i] + aux) \bmod 2$ 
6:      $aux = (A[i] + B[i] + aux) / 2$ 
7:    $C[i] = aux$ 
8:   return  $C$ 
```

6 Cormen 2.1-1

Illustrate the operation of INSERTION-SORT on the array A(31, 41, 59, 26, 41, 58)

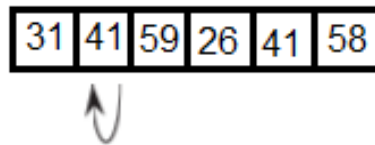


Figure 1: El 41 se mantiene en la misma posición

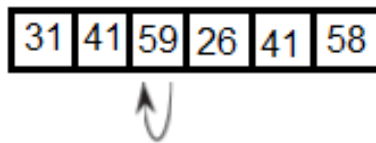


Figure 2: El 59 se mantiene en la misma posición

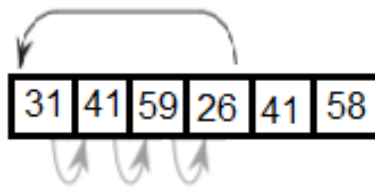


Figure 3: El 26 se debe desplazar al comienzo del arreglo y se mueven el 31, 41 y 59 una posición

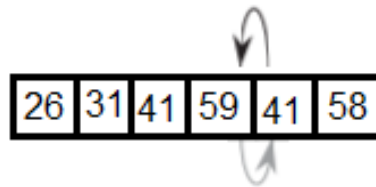


Figure 4: El 41 se intercambia de posición con el 59

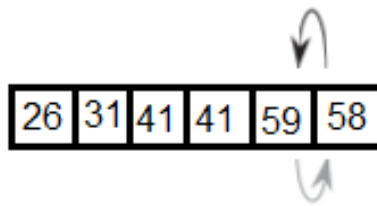


Figure 5: El 59 se intercambia de posición con el 58

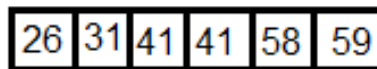


Figure 6: El arreglo está ordenado