

UNIVERSIDAD NACIONAL DE COLOMBIA
INGENIERIA DE SISTEMAS
ALGORITMOS 2007 - I
GRE COMPUTER SCIENCE

2. Which of the following decimal numbers has an exact representation in binary notation?

- (A) 0.1 (B) 0.2 (C) 0.3 (D) 0.4 **(E) 0.5**

Rta: / La respuesta es la (E). Porque $0.5 = \frac{1}{2} = 2^{-1}$ representación de pesos del número binario 0.1

3. Bob writes down a number between 1 and 1,000. Mary must identify that number by asking “yes/no” questions to Bob. Mary knows that Bob always tells the truth. If Mary uses an optimal algorithm, then she will determine the answer at the end of exactly how many questions in the worst case?

- (A) 1,000 (B) 999 (C) 500 (D) 32 **(E) 10**

Rta: / El algoritmo que Mary está utilizando es el de búsqueda binaria, con lo cual se generan divisiones sucesivas del intervalo entre dos:

$$\begin{aligned} \lceil 1000 / 2 \rceil &= \lceil 500 / 2 \rceil = \lceil 250 / 2 \rceil = \lceil 125 / 2 \rceil = \lceil 63 / 2 \rceil = \lceil 32 / 2 \rceil = \lceil 16 / 2 \rceil \\ &= \lceil 8 / 2 \rceil = \lceil 4 / 2 \rceil = \lceil 2 / 2 \rceil = 1 \end{aligned}$$

Encontrando 10 divisiones; ya que el algoritmo de búsqueda binaria es de orden $O = \log N$, se puede utilizar este método observando que el número de preguntas a realizar es: $\lceil \log_2 1000 \rceil = 10$

4. If x is a string, then x^R denotes the reversal of x . If x and y are strings, then $(xy)^R =$

- (A) xy^R (B) yx^R (C) $y^R x$ (D) $x^R y^R$ **(E) $y^R x^R$**

Rta: / Supongamos que $x = \text{“ABC”}$ $y = \text{“DEF”}$

Entonces $xy = \text{“ABCDEF”}$

Es decir, $(xy)^R = (\text{“ABCDEF”})^R$

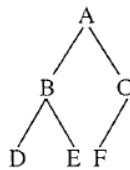
$= \text{“FEDCBA”}$

$= (\text{“FED”})(\text{“CBA”})$

Si $x^R = \text{“CBA”}$; $y^R = \text{“FED”}$

$(xy)^R = y^R x^R$

5. A procedure that printed the binary tree

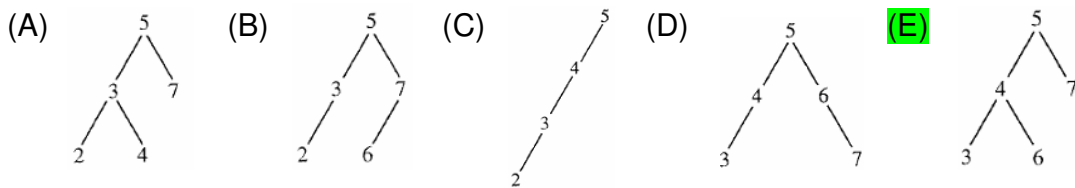


In postorden would produce as output

- (A) ABCDEF (B) ABDECF (C) DBEAFC (D) DEBFCA (E) DEFBCA

Rta: / El recorrido en postorden es: subárbol izquierdo, luego subárbol derecho y luego la raíz.

6. Which of the following is NOT a binary search tree?



Rta: / Los árboles de búsqueda binarios se caracterizan por que en la raíz se encuentra un valor clave el cual es mayor o igual que los valores del subárbol izquierdo y menor o igual que los valores del subárbol derecho. En el caso de (E) el 6 se encuentra en el subárbol izquierdo siendo mayor que 5 por lo cual no representa un árbol de búsqueda binaria.

7. Consider the following Pascal-like program fragment.

```

var i, j : integer;
procedure P(k, m: integer) ;
begin
    k := k - m ;
    m := k + m ;
    k := m - k ;
end ;
i := 2 ;
j := 3 ;
P(i, j) ;
  
```

If both parameters to P are passed by reference, what are the values of i and j at the end of the program fragment?

- (A) i = 0, j = 2 (B) i = 1, j = 5 (C) i = 2, j = 3 (D) i=3, j=2 (E) None of the above

Rta: /

```

k=i=2          m=j=3
k=k-m
=2-3
=-1
m=k+m
=(-1) +3
  
```

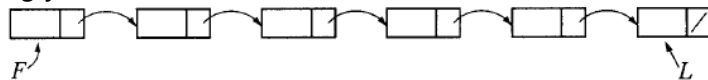
$$\begin{aligned}
&=2 \\
k &= m - k \\
&= 2 - (-1) \\
&= 3 \\
i &= k = 3 \qquad j = m = 2
\end{aligned}$$

8. A starvation-free job-scheduling policy guarantees that no job waits indefinitely for service. Which of the following job-scheduling policies is starvation-free?

- (A) Round-robin (B) Priority queuing (C) Shortest job first (D) Youngest job first (E) None of the above

Rta:/ En este tipo de planificación se le asigna a cada proceso un mismo tiempo de CPU determinado, teniendo en cuenta que el primero en llegar, es el primero en ser atendido. Si en ese tiempo el proceso no culmina vuelve a la cola de procesos para serle asignado otro tiempo.

9. Consider a singly linked list of the form



Where F is a pointer to the first element in the list and L is a pointer to the last element in the list. The time of which of the following operations depends on the length of the list?

- (A) Delete the last element of the list.
 (B) Delete the first element of the list.
 (C) Add an element after the last element of the list
 (D) Add an element before the first element of the list
 (E) Interchange the first two elements of the list

Rta: / para borrar el último elemento hay que colocar L apuntando al penultimo elemento de la lista. Para lo cual se debe recorrer la lista desde el principio.

10. $p := 1; k := 0;$
 while $k < n$ do
 begin
 $p := 2 * p;$
 $k := k + 1;$
 end ;

For the program fragment above involving integers p, k, and n, which of the following is a loop invariant; i.e., true at the beginning of each execution of the loop and at the completion of the loop?

- (A) $p = k + 1$ (B) $p = (k + 1)^2$ (C) $p = (k + 1)2^k$ (D) $p = 2^k$ (E) $p = 2^{k+1}$

Rta:/

p	k
1	0
2	1
4	2
8	3
...	...

Al observar los resultados de la prueba del algoritmo vemos que p es una potencia de 2 que esta dada por el valor de k.

Questions 13-14 relate to the following C-like program.

```
#include <stdio.h>
main()
{
    float sum = 0.0, j = 1.0 , i = 2.0 ;
    while( i / j > 0.001)
    {
        j = j + j ;
        sum = sum + i / j ;
        printf("%f \n", sum);
    }
}
```

13. How many lines of output does the program produce?

- (A) 0 – 9 (B) 10 – 19 (C) 20 - 29 (D) 30 – 31 (E) More than 39

Rta: /

i	j	i/j	sum
2	1	2	0
2	2	1	1
2	4	0,5	1,5
2	8	0,25	1,75
2	16	0,125	1,875
2	32	0,0625	1,9375
2	64	0,03125	1,96875
2	128	0,015625	1,984375
2	256	0,0078125	1,9921875
2	512	0,00390625	1,99609375
2	1024	0,001953125	1,99804688

Al observar los resultados de la prueba del programa vemos a partir de que j adquiere el valor de 2 sum empieza a mostrar los valores como salida, y muestra 10 valores antes de que i/j tome un valor inferior a 0.001.

14. Which of the following is the integer that best approximates the last number printed?

- (A) 0 (B) 1 **(C) 2** (D) 3 (E) 4

Tomando el último valor $\lceil 1,99804688 \rceil = 2$

15. An integer c is a common divisor of two integers x and y if and only if c is a divisor of x and c is a divisor of y . Which of the following sets of integers could possibly be the set of all common divisors of two integers?

- (A) $\{-6, -2, -1, 1, 2, 6\}$
(B) $\{-6, -2, -1, 0, 1, 2, 6\}$
(C) $\{-6, -3, -2, -1, 1, 2, 3, 6\}$
(D) $\{-6, -3, -2, -1, 0, 1, 2, 3, 6\}$
(E) $\{-6, -4, -3, -2, -1, 1, 2, 3, 4, 6\}$

Rta: / Tomando los valores extremos de cada intervalo sacamos los comunes divisores para estos valores, en el caso de (A) deben estar todos los divisores de 6 y -6, y no se encuentra el ± 3 , por lo cual (A), queda descartada; en el caso de (B) y (D) entre los divisores se encuentra el 0, ningún número es divisible por 0, con lo cual que descartadas estas dos opciones; en el caso de (E) entre los divisores aparece el ± 4 , pero entre los dos valores del intervalo no aparece ± 12 , con lo cual queda descartado (E); de tal manera que la única posibilidad de solución es (C) puesto que allí se encuentran todos los comunes divisores de ± 6 : $\pm 6, \pm 3, \pm 2, \pm 1$.

17. A particular parallel program computation requires 100 seconds when executed on a single processor. if 40 percent of this computation is "inherently sequential" (will not benefit from additional processors), then the theoretically best possible elapsed times for this program running with 2 and 4 processors, respectively, are

- (A) 20 and 10 seconds
(B) 30 and 15 seconds
(C) 50 and 25 seconds
(D) 70 and 55 seconds
(E) 80 and 70 seconds

Rta: /

Procesadores	Parte 1	Parte 2	Tiempo total
1	40	60	100
2	40	$60/2= 30$	70
4	40	$60/4= 15$	55

En todos los casos la primera parte es inherentemente secuencial, por cual se demorará 40 seg. La segunda parte puede ser ejecutada en paralelo por el número de procesadores disponibles, en el caso de 2 procesadores la velocidad aumenta el doble por lo tanto el tiempo disminuye, realizando el segundo

proceso en 30 seg. En el caso de 4 procesadores la velocidad se cuadruplica realizando el segundo proceso en 15 seg.

20. Let P be a procedure that for some inputs calls itself (i.e., is recursive). If P is guaranteed to terminate, which of the following statements must be true?
- I. P has a local variable.
 - II. P has an execution path where it does not call itself.
 - III. P either refers to a global variable or has at least one parameter.
- (A) I only.
(B) II only.
(C) I and II only.
(D) II and III only.
(E) I, II, and III.

Rta:/ Para garantizar que el procedimiento recursivo termine debe tener obligatoriamente un segmento donde no se llame a si mismo, de igual forma tener un parámetro que determina cuando termina la recurrencia o si se realiza el procedimiento y que debe ser global a la función, ya que si fuera local se redefiniría en cada recurrencia.

23. A particular disk unit uses a bit string to record the occupancy or vacancy of its tracks, with 0 denoting vacant and 1 denoting occupied. A 32-bit segment of this string has the hexadecimal value D4FE2003. The percentage of occupied tracks for the corresponding part of the disk, to the nearest percent, is
- (A) 12% (B) 25% (C) 38% **(D) 44%** (E) 62%

Rta: / D4FE2003= 1101|0100|1111|1110|0010|0000|0000|0011

Como hay 32-bit y solo están ocupados 14, el porcentaje que ocupa los espacios marcados con 1 es: $14 \cdot 100 / 32 = 43.75\% \approx 44\%$

26. Let S be the statement:
- for i:=1 to N do V[i]:= V[i] + 1
- Which of the following perform(s) the same changes to V as S?
- I. i:=0;
while i <=N do
begin i:= i+1; V[i]:=V[i]+1 end
 - II. i:=1;
while i < N do
begin V[i]:=V[i]+1; i:= i+1 end
 - III. i:=0;
while i < N do
begin V[i+1]:=V[i+1]+1; i:= i+1 end
- (A) I only
(B) II only
(C) III only

- (D) II and III only
- (E) I, II, and III

Rta:/ Tomando for($i=1; i \leq N; i++$), la opción I falla porque ejecuta una iteración de más con respecto al ciclo for, la opción II falla porque ejecuta una iteración menos, y por último la opción realiza las mismas iteraciones y el resultado es el mismo.

```
27. var i, j, x : integer ;
    read(x) ;
    i := 1 ; j := 1 ;
    while i < 10 do
    begin
        j := j * i ;
        i := i + 1 ;
        if i = x then exit
    end
```

for the program fragment above, which of the following statements about the variables i and j must be true after the execution of the fragment?

- (A) $(j = (x - 1)!) \wedge (i \geq x)$
- (B) $(j = 9!) \wedge (i = 10)$
- (C) $(j = 10!) \wedge (i = 10)$
- (D) $((j = 10!) \wedge (i = 10)) \vee ((j = (x - 1)!) \wedge (i = x))$
- (E) $((j = 9!) \wedge (i \geq 10)) \vee ((j = (x - 1)!) \wedge (i = x))$

Rta: / analizando el fragmento de programa podemos descartar las opciones (B) y (C), porque j no necesariamente tiene que ser $9!$ o $10!$ e i no tiene que ser obligatoriamente 10 , puesto que puede ser un valor menor dado por x ; la opción (A) se descarta porque i no puede ser mayor a x ; y por último la opción (D) se descarta porque j no llega a tomar el valor $10!$. En (E) si se ejecutan todas las iteraciones se da la primera parte, en caso de que se ejecuten menos se cumpliría la segunda condición.

Questions 29-30 are base on the following program heading and type declaration written in a Pascal-like language with garbage collection.

```
program Main(input, output);
type
    Link=↑Cell;
    Cell= record
        Info:integer;
        Next: Link
    end;
```

29. If the rest of the program is as follows,

```

var
  p1, p2 : Link ;
procedure A ;
  var
    p3 : Link ;
  begin
(ii)    new(p2) ;
(iii)   new(p3) ;
(iv)    new(p3↑.Next)
  end ;
begin (Main)
(i)     new(p1) ;
        A ;
(v)     ...
end.

```

Then, at line (v), which of the allocated Cells is (are) available for reclamation by the garbage collector?

- (A) None
- (B) Only the one allocated by (iii)
- (C) Only the ones allocated by (iii) and (iv)
- (D) Only the ones allocated by (ii), (iii) and (iv)
- (E) Those allocated by (i), (ii), (iii) and (iv)

30. In another program written in Pascal-like language with the same heading and type declaration, assume that choice must be made between the following two options.

Option *P*: the procedure *Push* defined by

```
procedure Push(n : integer ; h : Link) ;
var
  p : Link ;
begin
  new(p) ;
  p↑.Info := n ;
  p↑.Next := h ;
  h := p
end ;
```

to be used in the main program using a procedure call of the form

```
Push (n, Head) ;
```

Option *F*: the function *Push* defined by

```
function Push (n : integer ; h : Link) : Link ;
var
  p : Link ;
begin
  new(p) ;
  p↑.Info := n ;
  p↑.Next := h ;
  Push := p
end ;
```

to be used in the main program using an assignment of the form

```
Head := Push (n, Head) ;
```

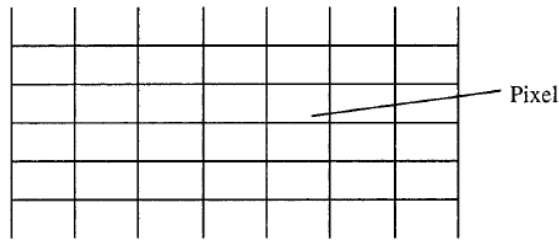
Suppose *Head* points to the first *Cell* of a list of *Cells*, and a new *Cell* is to be added at the front of the list, leaving *Head* pointing to the newly added *Cell*.

Which of the following implementations will FAIL to accomplish this objective?

- (A) Option *P* with parameters passed by value
- (B) Option *F* with parameters passed by value
- (C) Option *P* with parameters passed by reference
- (D) Option *F* with parameters passed by reference
- (E) Option *P* with parameters passed by name

Rta: / Detallando el programa podemos descartar la opción (E) porque Pascal elimino el paso de parámetros por nombre, permitiendo únicamente pasar parámetros por referencia y valor así que esta no es una razón para que no funcione el programa; asimismo podemos ver que en ninguno de los casos los parámetros se pasan por referencia porque no le anteponen la palabra VAR con lo cual podemos descartar las opciones (C) y (D) porque ninguno de los fragmentos lo usa; por último podemos descartar la opción (B) porque aunque los parámetros si son pasados por valor, se retorna las modificaciones hechas, con lo cual nos deja que la opción con la cual falla es la (A).

32. A black-and-white computer graphics display is divided up into an array of pixels as shown below.



Each of the pixels can take on one of eight grey levels ranging from 0 (white) to 7 (black). In order to prevent sharp discontinuities of shade, the software system that causes pictures to be displayed enforces the rule that the grey levels of two adjacent pixels cannot differ by more than two. How many of the 64 possible assignments of grey levels to two adjacent pixels satisfy this rule?

- (A) 24 (B) 32 (C) 34 (D) 40 (E) 64

Rta: / Un píxel tomado al azar puede tomar cualquiera de los 8 distintos niveles. Cualquiera de los píxeles adyacentes puede tomar alguno de los dos niveles superiores, o alguno de los dos niveles inferiores o el mismo nivel que el píxel inicial, de tal manera que este píxel adyacente puede tomar cualquiera de los 5 niveles. Teniendo en cuenta lo anterior, de las 64 posibles asignaciones, por medio del principio de multiplicación, las que satisfacen esta regla son $8 \cdot 5 = 40$.

33. A doubly linked list is declared as

```
Element = record
    Value      : integer ;
    Fwd, Bwd : ^Element
end ;
```

where *Fwd* and *Bwd* represent forward and backward links to adjacent elements of the list.

Which of the following segments of code deletes the element pointed to by *X* from the doubly linked list, if it is assumed that *X* points to neither the first nor the last element of the list?

- (A) $X \uparrow . Bwd \uparrow . Fwd := X \uparrow . Fwd ;$
 $X \uparrow . Fwd \uparrow . Bwd := X \uparrow . Bwd$
- (B) $X \uparrow . Bwd \uparrow . Fwd := X \uparrow . Bwd ;$
 $X \uparrow . Fwd \uparrow . Bwd := X \uparrow . Fwd$
- (C) $X \uparrow . Bwd \uparrow . Bwd := X \uparrow . Fwd ;$
 $X \uparrow . Fwd \uparrow . Fwd := X \uparrow . Bwd$
- (D) $X \uparrow . Bwd \uparrow . Bwd := X \uparrow . Bwd ;$
 $X \uparrow . Fwd \uparrow . Fwd := X \uparrow . Fwd$
- (E) $X \uparrow . Bwd := X \uparrow . Fwd ;$
 $X \uparrow . Fwd := X \uparrow . Bwd$
- (A) $X \uparrow . Bwd \uparrow . Fwd := X \uparrow . Fwd ;$
 $X \uparrow . Fwd \uparrow . Bwd := X \uparrow . Bwd ;$

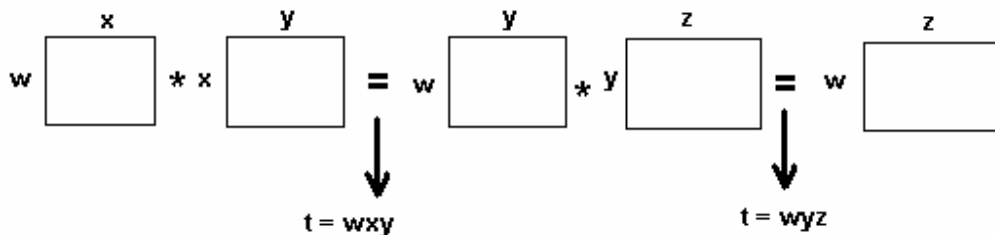
Rta:/ observando los segmentos de código se descarta en primer lugar la respuesta (E) porque lo que hacen las instrucciones es dejar apuntando los apuntadores al mismo al nodo donde esta X (Fwd ya no apunta al siguiente sino al de atrás y Bwd hace lo contrario); en el caso de (D) lo que se hace es dejar el Bwd del nodo anterior apuntando a su mismo Fwd, y deja el Fwd de su nodo anterior apuntando a su Bwd; en el caso de (C) crea una lista circular con los nodos anterior y posterior; y en el caso de (B) rompe la lista dejando el apuntador Fwd del nodo anterior apuntando a el mismo y el apuntador Bwd del nodo posterior apuntado a el mismo.

39. To compute the matrix product M_1M_2 , where M_1 has p rows and q columns and where M_2 has q rows and r columns, takes time proportional to pqr , and the result is a matrix of p rows and r columns. Consider the product of three matrices $N_1N_2N_3$ that have, respectively, w rows and x columns, x rows and y columns, and y rows and z columns. Under what condition will it take less time to compute the product as $(N_1N_2)N_3$ (i.e., multiply the first two matrices first) than to compute it as $N_1(N_2N_3)$?

- (A) There is no such condition; i.e., they will always take the same time.
 (B) $1/x + 1/z < 1/w + 1/y$
 (C) $x > y$
 (D) $1/w + 1/x < 1/y + 1/z$
 (E) $w + x > y + z$

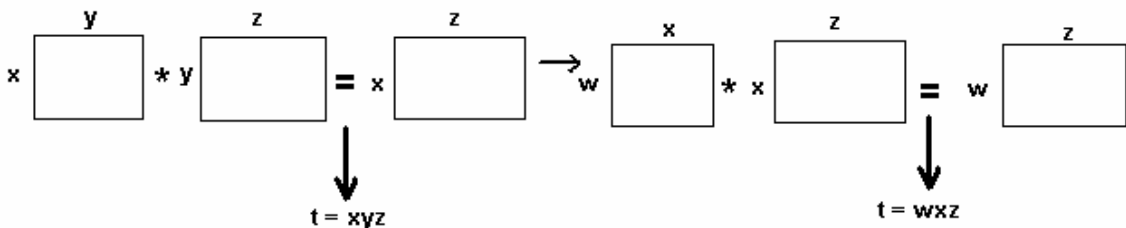
(B)

$(N_1N_2)N_3=$



$$t_{(N_1N_2)N_3} = wxy + wyz$$

$N_1(N_2N_3)=$



$$t_{N_1(N_2N_3)} = xyz + wxz$$

$$\begin{aligned}
& wxy + wyz < xyz + wxz \\
\Rightarrow & wy(x + z) < xz(y + w) \\
\Rightarrow & \frac{(x + z)}{xz} < \frac{(y + w)}{wy} \\
\Rightarrow & 1/x + 1/z < 1/w + 1/y
\end{aligned}$$

41. Let A be a finite nonempty set with cardinality n . The number of subsets $S \subseteq A$ having odd cardinality is
- (A) n
 - (B) $2^{n/2}$
 - (C) 2^{n-1}
 - (D) 2^n
 - (E) not determinable except in terms of whether n is even or odd
42. A certain algorithm A has been shown to have running time $O(N^{2.5})$, where N is the size of the input. Which of the following is NOT true about algorithm A ?
- (A) There exist constants C_1 and C_2 such that for all N the running time is less than $C_1N^{2.5} + C_2$ seconds.
 - (B) For all N , there may be some inputs for which the running time is less than $N^{2.4}$ seconds.
 - (C) For all N , there may be some inputs for which the running time is less than $N^{2.6}$ seconds.
 - (D) For all N , there may be some inputs for which the running time is more than $N^{2.4}$ seconds.
 - (E) For all N , there may be some inputs for which the running time is more than $N^{2.6}$ seconds.

Rta:/ la notación O muestra una cota superior asintótica, es decir que el algoritmo en peor de los casos es $O(N^{2.5})$ por lo tanto puede haber tiempos de ejecución, según la entrada, menores o iguales a $N^{2.5}$ por lo tanto la única respuesta que contradice este aspecto es la (E).

43. A data structure is comprised of nodes each of which has exactly two pointers to other nodes, with no null pointers. The following C program is to be used to count the number of nodes accessible from a given node. It uses a mark field, assumed to be initially zero for all nodes. There is a statement missing from this code.

```

struct test {int info, mark; struct test *p, *q; }
int nodecount (struct test *a)
{
    if (a->mark) return 0 ;
    return nodecount (a->p) + nodecount (a->q) + 1 ;
}

```

- Which changes should be made to make the program work properly?
- (A) Add "a->mark=1;" as the first statement.

(B) Add "a->mark=1;" after the "if" statement.

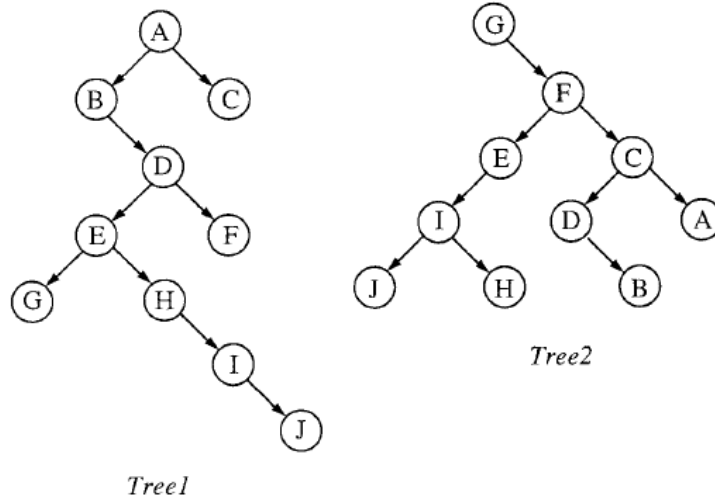
(C) Add "a->mark=1;" as the last statement.

(D) Add "a->mark=0;" after the "if" statement.

(E) Add "a->mark=0;" as the last statement.

Rta:./ observando el programa, si usáramos la opción (A) hallaría un solo nodo y puesto que el if esta después retornaría un 0, enviando un número erróneo de nodos y ni siquiera teniendo en cuenta el único nodo encontrado; en el caso de la opción (C) y (E) no encontraría nunca una opción de terminación en la llamada recursiva puesto que la instrucción esta luego del return; en el caso de (D) haría llamadas recursivas infinitamente porque nunca mark va a adoptar un valor distinto de 0; por lo tanto la única opción restante es la (B) puesto que va marcando cada nodo visitado y hace el retorno cuando encuentra de nuevo el nodo inicial, este programa se podría usar en el conteo de nodos de una lista circular.

47. If the *Tree1* and *Tree2* are the trees indicated below,



Which traversals of *Tree1* and *Tree2*, respectively, will produce the same sequence of node names?

(A) Preorder, postorder

(B) Postorder, inorder

(C) Postorder, postorder

(D) Inorder, inorder

(E) Postorder, preorder

Rta:./ Posorden (IDR) *Tree1*: GJIHEFDBCA

Inorden (IRD) *Tree2*: GJIHEFDBCA

48. Let *A* be a finite set with *m* elements, and let *B* be a finite set with *n* elements. The number of distinct functions mapping *A* into *B* is

(A) n^m (B) $n!/(n-m)!$ (C) $n!$ (D) $n!/(m!(n-m)!)$ (E) 2^{n+m}

58. Of the following, which gives the best upper bound for the value of $f(N)$ where f is a solution to the recurrence

$$f(2N + 1) = f(2N) = f(N) + \log N \text{ for } N \geq 1,$$

With $f(1)=0$?

- (A) $O(\log N)$ (B) $O(N \log N)$ (C) $O(\log N) + O(1)$
 (D) $O((\log N)^2)$ (E) $O(N)$

Rta: / $f(2N+1) = f(2N) + f(1) = f(2N) + 0 = f(2N) = f(N+N) = f(N) + \log N$
 $\Rightarrow f(N) = \log N$
 $\Rightarrow f(N) + \log N = \log N + \log N = 2 \log N \Rightarrow O(\log N)$

Questions 63-64 refer to the following information:

An array $A[1, 2, \dots, n]$ is said to be k -ordered if

$$A[i-k] \leq A[i] \leq A[i+k]$$

for each i such that $k < i \leq n - k$. For example, the array 1 4 2 6 3 7 5 8 is 2-ordered.

63. In a 2-ordered array of $2N$ elements, what is the maximum number of positions that an element can be from its position if the array were 1-ordered?
 (A) $2N - 1$ (B) 2 (C) $N / 2$ (D) 1 (E) N

Rta: /

Si $N=8$, entonces $2N=16$

Si tomamos un valor intermedio a 16 en el arreglo 1-ordered como el 8 o el 9, siendo los peores casos, podemos ver que el máximo número de posiciones que cualquier número puede estar de su posición inicial es N , que en este caso es 8

2-ordered

1	9	2	10	3	11	4	12	5	13	6	14	7	15	8	16
							<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	
	<u>8</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

1-ordered

64. In an array of $2N$ elements that are both 2-and 3-ordered, what is the maximum number of that an element can be from its position if the array were 1-ordered?
 (A) $2N - 1$ (B) 2 (C) $N / 2$ (D) 1 (E) N

Diana Carolina García Cortés 257230
 Daniel Ricardo Forero Cárdenas 257278
 Leandro Liu Weng 257245
 Víctor David Ortiz Ortiz 257256