Estructuras de Datos 1 - ST0245

Examen Parcial 2

Nombre: ..............................................................

Departamento de Informática y Sistemas

Universidad EAFIT

**CONCEPTUALIZACION**

1. (10 %) Sea A un **árbol binario de búsqueda** al que se le ingresan los siguientes elementos en su orden: 12, 3, 5, 8, 2, 1, 9, 17. Podríamos afirmar que el árbol recorrería n nodos antes de encontrar el número 8. ¿Cuál debe ser el valor de n?
2. 2
3. 1
4. 3
5. 5
6. (10 %) Para insertar n elementos (1 <= n <= 1000) en un árbol binario de búsqueda de forma que el código **no sea recursivo**, ¿Cuál sería la mejor solución?
7. Hago un código con múltiples condicionales analizando tanto el hijo izquierdo como el hijo derecho.
8. Es imposible insertar un elemento en un Árbol Binario de Búsqueda si recursión.
9. Con un solo ciclo.
10. Con dos ciclos anidados. El de afuera para analizar los hijos de la izquierda y el de adentro para analizar los hijos de la derecha.
11. (1 0%) ¿Cuál es la complejidad asintótica para insertar un elemento en un árbol binario de búsqueda?
12. O (n)
13. O (n log n)
14. O (log máximo (altura Árbol, elemento\_Insertar))
15. O (log n)

**BACKTRACKING**

1. (40 %)Dayla y Kefo están aquí de nuevo; y como ya sabemos están obsesionados con los números. En esta vez han traído un juego muy interesante, en el cual Kefo en primer lugar escoge un numero n (1 ≤ n ≤ 20) y en segundo lugar escoge tres números a, b y c (1 ≤ a ≤ 9, 1 ≤ b ≤ 9, 1 ≤ c ≤ 9). Después Kefo le entrega estos números a Dayla y él (Dayla) le tiene que decir a Kefo la cantidad máxima de números tomados de a, b y c (posiblemente se puede tomar un número más de una vez) que al sumarlos dan el valor n.

Supongamos que n = 14 y a = 3, b = 2, c = 7. Ahora miremos las posibilidades de obtener 14 con a, b y c.

7 + 7 = 14 cantidad = 2

7 + 3 + 2 + 2 = 14 cantidad = 4

3 + 3 + 3 + 3 + 2 = 14 cantidad = 5

… (Otras posibles combinaciones).

2 + 2 + 2 + 2 + 2 + 2 + 2 = 14 cantidad = 7

Entonces podríamos demostrar que la cantidad máxima de números es 7, lo cual sería la respuesta que da Dayla a Kefo.

Como Dayla es muy astuto ha diseñado un algoritmo para determinar la cantidad máxima de números y quiere que le ayudes a terminar su código. Ayuda a Dayla a terminar su código.

public int solucionar (int n, int a, int b, int c){

//Caso base

if((1)\_\_\_\_\_\_\_){ (15 %)

return 0;

}

//Primero calculemos la respuesta con un valor cualesquiera.

int res =(2) \_\_\_\_\_\_\_\_\_\_\_; ( 15 %)

res = Math.max((3)\_\_\_\_\_\_,\_\_\_\_\_\_\_\_\_\_); (5 %)

res = Math.max((4)\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_); (5 %)

return res;

}

**ARBOLES BINARIOS**

1. (10 %)Dayla y Kefo están enojados uno con el otro y Kefo ha dañado a Dayla su código para determinar cuál es la altura de un árbol binario. Ahora Dayla no recuerda como lo había hecho y te pide ayuda para que lo completes.

Int altura(Nodo raíz){

// Caso base

If(raíz == null)

return 0;

Int izq =\_\_\_\_\_\_\_ (5 %);

Int der =\_\_\_\_\_\_; (5 %)

return max(izq, der);

}

1. (10 %)Sea A un **árbol binario de búsqueda** al cual se le van a insertar los siguientes elementos numéricos.

7, 9, 4, 13, 15, 18, 23, 12, 78, 34, 12, 9, 1, 5, 2.

Después de haber insertado los elementos anteriores y haber hecho un recorrido in-orden del árbol cual sería la salida del recorrido in-orden.

1. 1, 2, 4, 5, 7, 9, 12, 13, 15, 18, 23, 34, 78.
2. 2, 1, 5, 4, 12, 34, 78, 23, 18, 15, 13, 9, 7.
3. 7, 4, 1, 2, 5, 9, 13, 12, 15, 18, 23, 78, 34.
4. 13, 15, 18, 23, 34, 78, 1, 2, 4, 5, 7, 9, 12.
5. (10 %)Hoy es el último día de Dayla y Kefo en el Colegio y por diversas situaciones hoy presentan su último parcial de Matemáticas. Casi todos los puntos son simples excepto por uno. Vamos a hablar de ese punto. Supongamos que nos entregan una función **F(x) unimodal** (siempre creciente o siempre decreciente) en un intervalo **I: [l, r],** la idea es encontrar el máximo y mínimo de esa función en el intervalo **I**. Como el profesor permitió abrir el computador en clase durante los últimos 15 minutos Dayla y Kefo aprovecharon y utilizaron su código de **ternary search** para determinar tanto el máximo como el mínimo. Sin embargo, ambos se dieron cuenta que su código no andaba bien y te lo entregan a ti para que lo corrijas. ¿Podrías ayudar a Dayla y a Kefo a corregir su código?

//Esta es la función a evaluar y el código no está malo.

public int F(int x){

return 2 \* x \* x + 7 – 12 \* x;

}

//Este es el ternary search y la función está mala.

public double buscarMin(int s, int e){

double l = s;

double r = e;

for(int i = 0;i<2\*10\*11;i++){

double l1 = (l\*3+r )/2;

double l2 = (l+2\*r)/3;

if(F(l1) < F(l2)){

r = l2;

l = l1;

}

}

double x = l;

return F(x);

}

public double buscarMax(int s, int e){

double l = s;

double r = e;

for(int i = 0;i<2\*10\*11;i++){

double l1 = (l\*3+r )/2;

double l2 = (l+2\*r)/3;

if(F(l1) > F(l2)){

r = l2;

l = l1;

}

}

double x = l;

return F(x);

}

¿Cuál es el error en el código de Dayla y Kefo?

1. El error se soluciona cambiando double l1 = (l\*3+r )/2; por

doublé l1 = (l \* 2 + r) / 2;

1. El error se soluciona cambiando double l2 = (l+2\*r)/3; por

double l1 = (l+3+r )/3;

1. El error se soluciona cambiando double l1 = (l\*3+r )/2; por

double l1 = (l+3+r )/2;