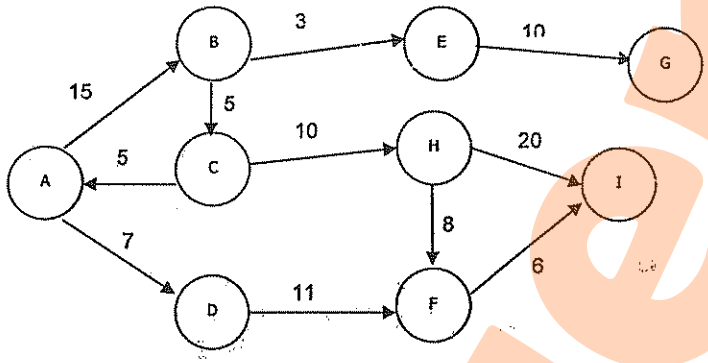
UNLP. Facultad de Informática. Nombre y apellido: .....
17/03/2011 Nro. Alumno: .....

Algoritmos y Estructuras de Datos
Curso de Verano 2011

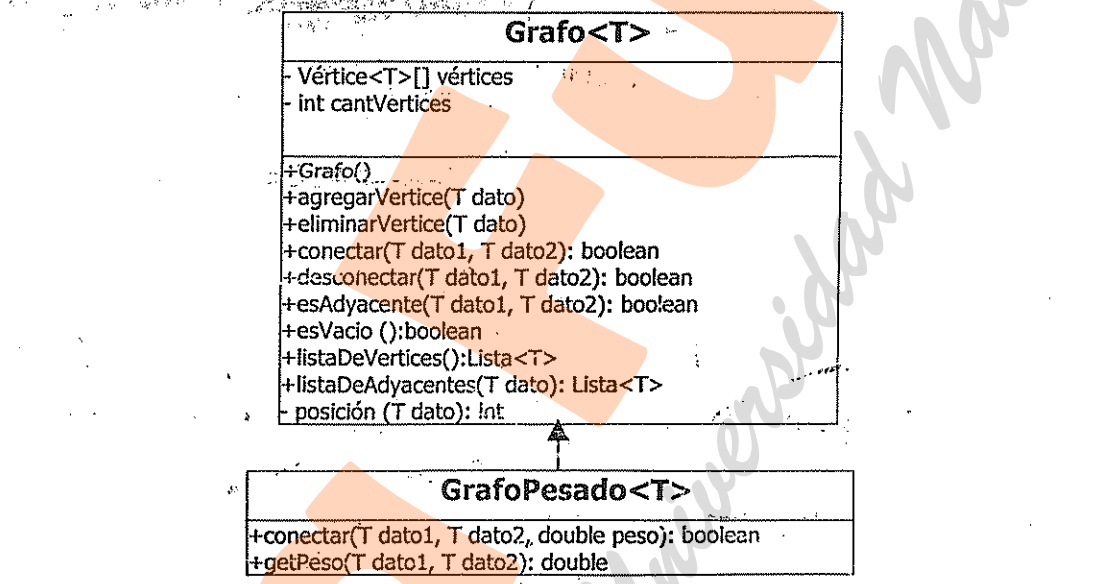
Recuperatorio del TEMA GRAFOS

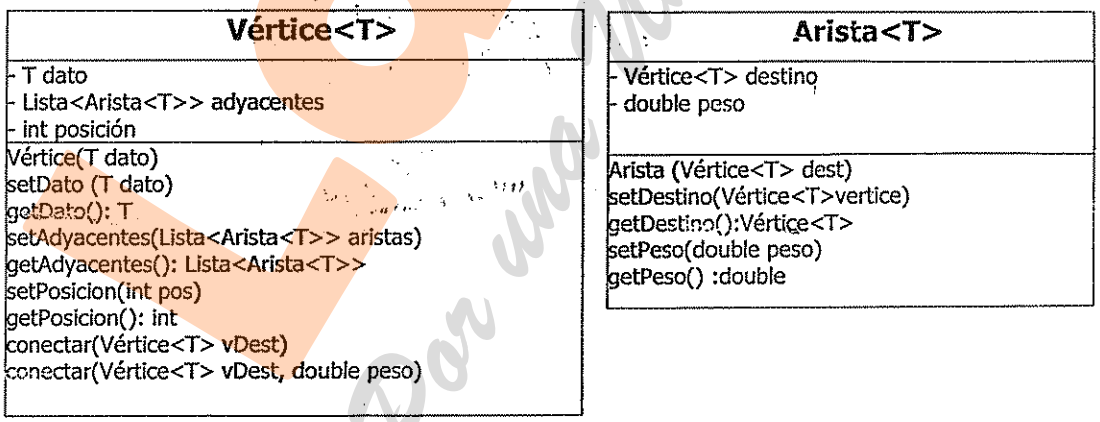
Dado un grafo orientado y pesado positivamente, y ***dos*** nodos de dicho grafo, implementar un método que encuentre y muestre (si existe) un camino simple de exactamente costo total ***c*** (***c > 0***).
Recordar que un *camino simple* es aquél en el que todos los nodos son distintos (excepto el primero y el último que pueden ser iguales)

Ejemplos :
1.- ***C = 50*** y los nodos ***A*** e ***I*** --- El camino simple de costo 40 es : A – B – C – H – I
2.- ***C = 25*** y los nodos ***A*** e ***I*** --- No existe el camino simple de costo 25 entre los nodos A e I
3.- ***C = 28*** y los nodos ***A*** y ***E*** --- No existe el camino simple de costo 28 entre los nodos A y E



Utilice las siguientes operaciones sin necesidad de implementarlas.





Algoritmos y Estructuras de Datos - Curso 2013
Parcial – 1era Fecha
Sábado 29 de Junio

Ejercicio 1.

Dada el siguiente algoritmo,

a.- Calcular su T(n), detallando los pasos seguidos para llegar al resultado.

b.- Calcular su O(n) justificando usando la definición de big-OH.

```
public static void uno ( int n) {

  for (int i = 1;  i <= n;  i ++) {

     for (int j = 1;  j<= i;  j++)
         algo_de_O(1);
  }
  If (n>1)
      for (int i = 1;  i <= 4;  i++)
          uno (n div 2);

}
```

En el caso de ser necesario tenga presente las siguientes series:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \qquad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$$

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \qquad \sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}$$

Ejercicio 2.

Llamaremos a un árbol general creciente si en cada nivel del árbol la cantidad de nodos que hay en ese nivel es igual al valor del nivel mas 1; es decir, el nivel 0 tiene exactamente un nodo, el nivel 1 tiene exactamente dos nodos, el nivel k tiene exactamente k+1 nodos. Comprobar si un árbol general es creciente y en caso que lo sea retornar el nodo del árbol con mayor cantidad de hijos. Sino lo es retornar null

Ejercicio 3.

Es sólo una moda o vino para quedarse? No se sabe, pero como parte de un plan fuerte de expansión en Argentina el número de locales abiertos de una muy conocida cadena internacional de café ha aumentado 50% en la primera mitad de 2013 con respecto al mismo período del año 2012.

Al parecer las personas se han vuelto tan adictos a estas tiendas de café gourmet que los inmuebles que están cerca de estas cafeterías obtienen mejores rentas. Esto ha sido notado por una compañía de bienes raíces, que está interesada en identificar si una determinada esquina es un lugar valioso en términos de su proximidad al mayor números de cafés de esta cadena.

Para ello cuentan con un mapa de la ciudad representada en un grafo donde, cada arista indica la cantidad de cafés en esa cuadra. Supongamos que una persona promedio está dispuesta a caminar un número de cuadras fijo X para obtener su café matinal. Usted tiene que determinar si la esquina en cuestión es valiosa. Es considerada valiosa si el número de cafés en los que una persona puede alcanzar es mayor a un número fijo arbitrario Y.

Algoritmos y Estructuras de Datos - Curso 2013
Viernes 12 de Julio

Ejercicio 1.

Dada el siguiente algoritmo,

a.- Calcular su T(n), detallando los pasos seguidos para llegar al resultado.

b.- Calcular su O(n) justificando usando la definición de big-OH.

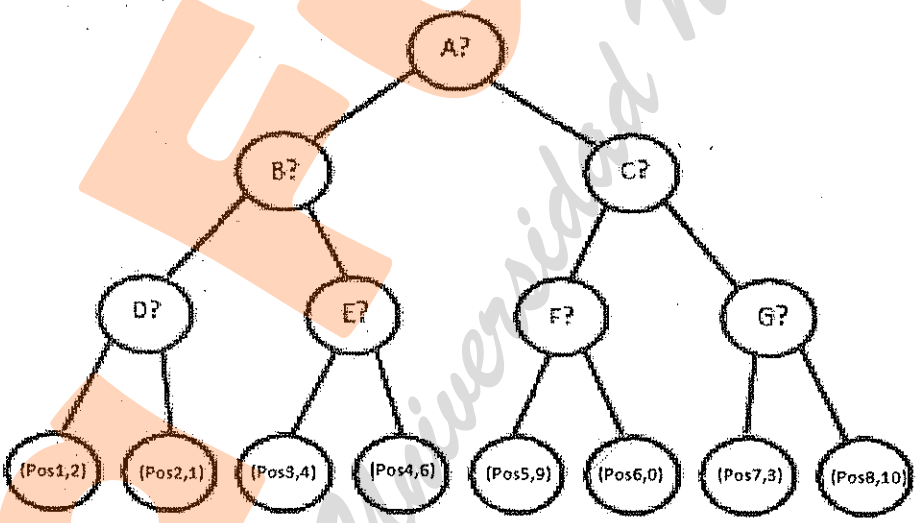
```
public static void main(String[] args) {
    int sum = 0, sum1 = 0, sum2 = 0;
    for (double i = 1; i <= n; i+=0.5) {
        ++sum1;
        for (double j = 1; j <= sum1; j++) {
            sum2++;
            for (double k = j; k <= sum1; k++)) {
                sum += sum1 + sum2;
            }
        }
    }
}
```

Ejercicio 2.

Un árbol de disyunción binario ponderado es un árbol binario que encadena una serie de preguntas y las posibles alternativas. Así en un árbol de este tipo, los nodos que no son hojas representan disyuntivas y las aristas son las posibles alternativas que conectan con otras preguntas. Por otro lado, las hojas representan los resultados posibles a los cuales se puede arribar. Dichos resultados tienen asociados un coeficiente comprendido en el intervalo [0,10], donde 0 representa el peor resultado posible y 10 el mejor resultado esperable.

Implementar un método que dado un camino que comienza en la raiz y *no* llega a una hoja, retorne los 3 mejores resultados esperables. (El camino de decisiones pasado como parámetro existe en el árbol y es uno tal que siempre permite retornar los 3 mejores resultados)

Ejemplo: Dado el camino A?,C? el método debe devolver Pos8,Pos5,Pos7.



Ejercicio 3.

La Agencia de Control de Enfermedades Infecciosas de la Argentina (ACEIA) necesita generar distintas proyecciones computacionales relacionadas con la propagación de enfermedades infecciosas en nuestro país. La ACEIA trabaja bajo la hipótesis de que cualquier tipo de epidemia infecciosa tendrá como punto de origen de la enfermedad (POE) a ciudades limítrofes del país o ciudades que cuenten con aeropuertos.

La ACEIA cuenta con un mapa epidemiológico para realizar sus proyecciones. El mismo contiene las ciudades de todo el país y los caminos que comunican dos ciudades.

La ACEIA le provee un grafo que modela la información del mapa y lo ha contratado a Ud. para armar la siguiente proyección:

Todas las ciudades afectadas en la propagación de una enfermedad dentro de un periodo de ventana de cinco días. La enfermedad se inicia en una ciudad indicada y al día siguiente infectará a todas las ciudades adyacentes. Al otro día a las adyacentes de las adyacentes y así sucesivamente hasta alcanzar el período de ventana.

ORIGINAL

Algoritmos y Estructuras de Datos - Curso 2011
3er Parcial - Jueves 11 de Agosto

Ejercicio 1.

Dada la siguiente recurrencia,

a.- Calcular el T(n) resolviendo la recurrencia, detallando los pasos seguidos para llegar al resultado.

b.- Calcular el O(n) justificando usando la definición de big-OH.

$$T(n) \begin{cases} (4/5)C\log_5(n) & n = 1 \\ (Cn)/5 + 5T(n/5) & n > 1 \end{cases}$$

Ejercicio 2.

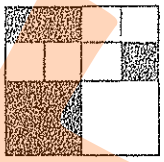
Un *quadtree* es una representación usada para cubrir un espacio cuadrado en dos dimensiones y posteriormente utilizado para determinar ciertas condiciones entre objetos en el mismo.

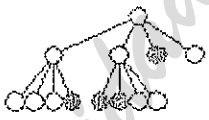
Un artista moderno trabaja con imágenes codificadas en *quadtrees*. El *quadtree* es un árbol 4-ario que codifica a una imagen con el siguiente criterio:

- Si toda la imagen tiene un mismo color, la misma es representada por un único nodo que almacene un dato que represente a ese color.

- En caso contrario, se divide la imagen en cuatro cuadrantes que se representan en el árbol como un nodo con 4 hijos, y cada hijo es la conversión de cada una de las partes de la imagen.

El artista desea saber cuántos pixeles de color negro posee una imagen dada. Usted debe implementar un método, que dado un *quadtree* y una cantidad total de pixeles, cuente cuantos pixeles de color negro contiene la imagen codificada en él.





Para el quadtree de la Figura, la salida del método sería 448

La figura muestra un ejemplo del árbol *quadTree* correspondiente a la imagen de la izquierda de 32 x 32 pixeles (1024 pixeles en total). Cada nodo en un *quadtree* es una hoja o tiene 4 hijos. Los nodos se recorren en sentido contrario a las agujas del reloj, desde la esquina superior derecha.

Ejercicio 3.

Una agencia de viajes publicita un conjunto de promociones turísticas, las cuales son totalmente configurables por los clientes de dicha agencia. Las promociones tienen un precio en pesos por tantos km, así un cliente que compre un paquete aéreo de x km podrá ir a los destinos que se encuentran a esa distancia.

La agencia desea tener en tiempo y forma todos los destinos posibles desde todos los orígenes posibles para un kilometraje dado y saber todas las escalas que deberá hacer el avión desde el origen hasta llegar al destino. Para ello contrató un programador JAVA que modele la situación antes descripta con un dígrafo pesado positivamente, donde los vértices del mismo corresponden a las ciudades y las aristas las distancias en kilómetros entre las ciudades que conecta.

Parcial Teórico de AyED para promoción 1º fecha – 2013

1) Explique cómo es el algoritmo BuildHeap. Utilícelo para construir una maxHeap con las siguientes claves. (0.5 puntos)
20, 4, 5, 37, 16, 45, 3, 64, 51, 100, 50, 15, 6, 10

a) ¿De qué orden es el algoritmo BuildHeap? Justifique. (0.5 puntos)

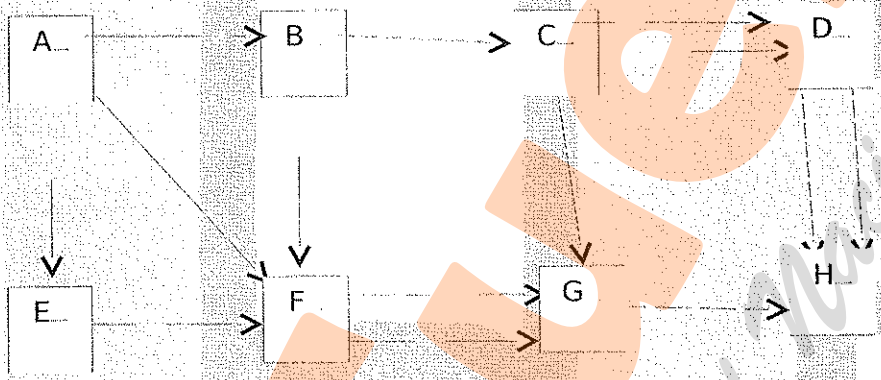
2) Dado un árbol general de grado K y altura H.

a) ¿Cantidad máxima de nodos en un árbol lleno? (0.5 puntos)
b) ¿Cantidad máxima y mínima de nodos en un árbol completo? (0.5 puntos)

3) Se considera una variación del algoritmo MergeSort, en el cual se divide la lista a ser ordenada en 3 sublistas del mismo tamaño, cada sublista se ordena recursivamente y luego se mezclan las 3 listas para obtener la lista final ordenada. Exprese y resuelva la función del tiempo de ejecución del algoritmo descripto. (2 puntos)

4) Una solución clásica para determinar las componentes fuertemente conexas de un grafo dirigido "G" es usar dos recorridos en profundidad (DFS). El primero determina el orden en que los nodos serán considerados en el siguiente paso. El segundo recorrido trabaja con el grafo traspuesto y construye un bosque donde cada árbol representa una componente conexa. El grafo transpuesto "GT" puede construirse invirtiendo todas las aristas del grafo original.

a- indique la numeración de los vértices que surge a partir del recorrido DFS (Sobre el dibujo) (1 punto)
b- Indique las componentes fuertemente conexas del siguiente grafo (0.5 puntos)



5)

```
public abstract class Test{
        public abstract void métodoA();
        public abstract void metodoB()
        {
                System.out.println("Hola");
        }
}
```

a- ¿Qué cambios (hechos independientemente) permitirían que el código compile? (indique 2 opciones) (0.5 puntos)
b- ¿Qué cambios haría en la clase para transformarla en una interface? ¿Qué beneficios tiene declarar una interface respecto de la clase? (0.5 puntos)

6) Considérese la información registrada en una agencia de viajes referente a vuelos existentes entre todos los aeropuertos del mundo. Supongamos que toda conexión aérea es ida y vuelta. Queremos resolver el siguiente problema, dados los aeropuertos A y B ¿Cuál es el mínimo numero de transbordos en un viaje aéreo de A hasta B?

a- De los algoritmos de recorrido de grafos conocidos, ¿cuál considera más eficiente para este problema? Explíquelo. Exprese el orden de ejecución. Justifique (1.5 puntos)
b- ¿Conoce otros algoritmos que resuelvan este problema? ¿Por qué no eligió ese en el punto a? (1.5 puntos)

Algoritmos y Estructuras de Datos - Curso 2013
Parcial -- 1era Fecha
Sábado 29 de Junio

Ejercicio 1.

Dada el siguiente algoritmo,

a.- Calcular su T(n), detallando los pasos seguidos para llegar al resultado.

b.- Calcular su O(n) justificando usando la definición de big-OH.

```
public static void uno ( int n) {

   for (int i = 1;  i <= n;  i ++) {

      for (int j = 1;  j<= i;  j++)
          algo_de_O(1);
   }
   If (n>1)
       for (int i = 1;  i <= 4;  i++)
           uno (n div 2);

}
```

En el caso de ser necesario tenga presente las siguientes series:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}, \quad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}, \quad \sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$$

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \qquad \sum_{i=0}^{n} c^i = \frac{c^{n+1} - 1}{c - 1}$$

Ejercicio 2.

Llamaremos a un árbol general **creciente** si en cada nivel del árbol la cantidad de nodos que hay en ese nivel es igual al valor del nivel mas 1; es decir, el nivel 0 tiene exactamente un nodo, el nivel 1 tiene exactamente dos nodos, el nivel k tiene exactamente k+1 nodos. Comprobar si un árbol general es creciente y en caso que lo sea retornar el nodo del árbol con mayor cantidad de hijos. Sino lo es retornar null

Ejercicio 3.

Es sólo una moda o vino para quedarse? No se sabe, pero como parte de un plan fuerte de expansión en Argentina el número de locales abiertos de una muy conocida cadena internacional de café ha aumentado 30% en la primera mitad de 2013 con respecto al mismo período del año 2012.

Al parecer las personas se han vuelto tan adictos a estas tiendas de café gourmet que los inmuebles que están cerca de estas cafeterías obtienen mejores rentas. Esto ha sido notado por una compañía de bienes raíces, que está interesada en identificar si una determinada esquina es un lugar valioso en términos de su proximidad al mayor números de cafés de esta cadena.

Para ello cuentan con un mapa de la ciudad representada en un grafo donde, cada arista indica la cantidad de cafés en esa cuadra. Supongamos que una persona promedio está dispuesta a caminar un número de cuadras fijo X para obtener su café matinal. Usted tiene que determinar si la esquina en cuestión es **valiosa**. Es considerada **valiosa** si el número de cafés en los que una persona puede alcanzar es mayor a un número fijo arbitrario Y.

Algoritmos y Estructuras de Datos - Curso 2013
Viernes 12 de Julio

Ejercicio 1.

Dada el siguiente algoritmo,

a.- Calcular su T(n), detallando los pasos seguidos para llegar al resultado.

b.- Calcular su O(n) justificando usando la definición de big-OH.

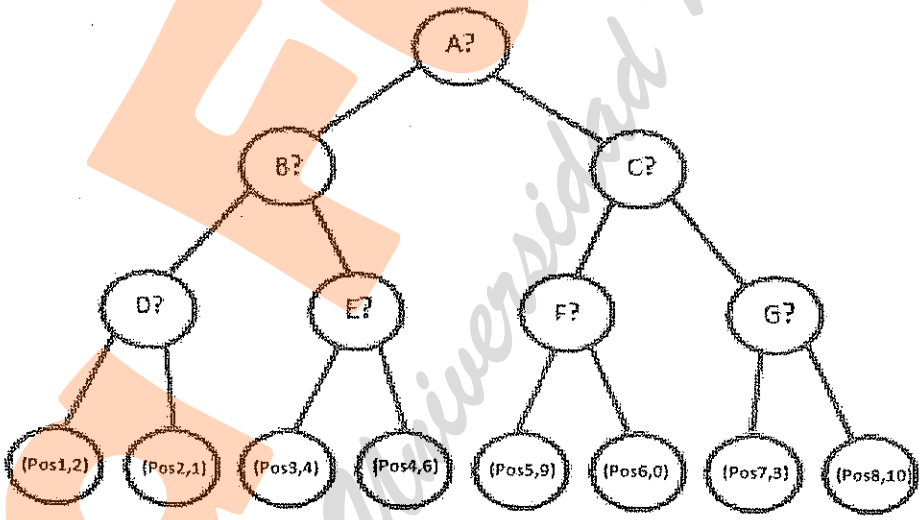
```
public static void main(String[] args) {
    int sum = 0, sum1 = 0, sum2 = 0;
    for (double i = 1; i <= n; i+=0.5) {
        ++sum1;
        for (double j = 1; j <= sum1; j++) {
            sum2++;
            for (double k = j; k <= sum1; k++)) {
                sum += sum1 + sum2;
            }
        }
    }
}
```

Ejercicio 2.

Un árbol de disyunción binario ponderado es un árbol binario que encadena una serie de preguntas y las posibles alternativas. Así en un árbol de este tipo, los nodos que no son hojas representan disyuntivas y las aristas son las posibles alternativas que conectan con otras preguntas. Por otro lado, las hojas representan los resultados posibles a los cuales se puede arribar. Dichos resultados tienen asociados un coeficiente comprendido en el intervalo [0,10], donde 0 representa el peor resultado posible y 10 el mejor resultado esperable.

Implementar un método que dado un camino que comienza en la raiz y *no* llega a una hoja, retorne los 3 mejores resultados esperables. (El camino de decisiones pasado como parámetro existe en el árbol y es uno tal que siempre permite retornar los 3 mejores resultados)

Ejemplo: Dado el camino A?,C? el método debe devolver Pos8,Pos5,Pos7.



Ejercicio 3.

La Agencia de Control de Enfermedades Infecciosas de la Argentina (ACEIA) necesita generar distintas proyecciones computacionales relacionadas con la propagación de enfermedades infecciosas en nuestro país. La ACEIA trabaja bajo la hipótesis de que cualquier tipo de epidemia infecciosa tendrá como punto de origen de la enfermedad (POE) a ciudades limítrofes del país o ciudades que cuenten con aeropuertos.

La ACEIA cuenta con un mapa epidemiológico para realizar sus proyecciones. El mismo contiene las ciudades de todo el país y los caminos que comunican dos ciudades.

La ACEIA le provee un grafo que modela la información del mapa y lo ha contratado a Ud. para armar la siguiente proyección:

Todas las ciudades afectadas en la propagación de una enfermedad dentro de un periodo de ventana de cinco días. La enfermedad se inicia en una ciudad indicada y al día siguiente infectará a todas las ciudades adyacentes. Al otro día a las adyacentes de las adyacentes y así sucesivamente hasta alcanzar el período de ventana.

Algoritmos y Estructuras de Datos - Redictado 2013
Sábado 19 de Octubre

Willy es una abeja macho y Maya su mejor amiga (que por cierto, es una abejita que le gusta ayudar a sus amigos). Willy ha descubierto que no tiene padre y está preocupado por esto.

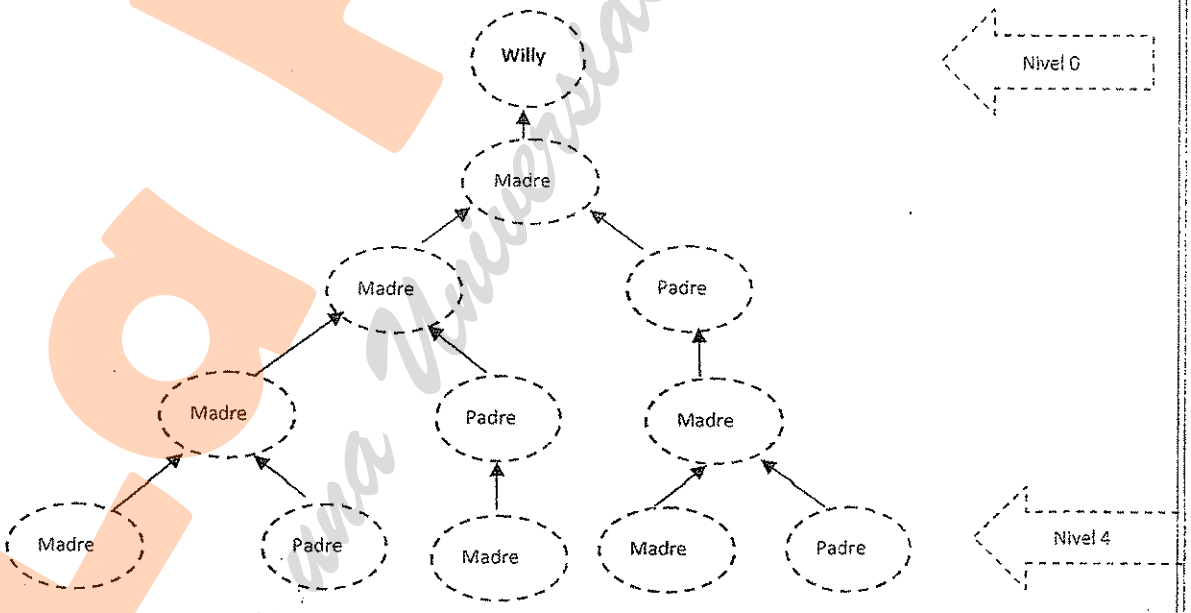
Maya conoce que las féminas tienen 2 padres (un padre y una madre), pero los machos tienen únicamente madre (no padre). Esto se debe a que si un huevo es puesto por una hembra sin pareja, del cascarón nace un macho. Por el contrario, si el huevo es fertilizado por un macho, nace una fémina.

Después de que Maya habla con Willy acerca de este tema, él comienza a pensar el número de ancestros que él tiene. Él tiene una madre, dos abuelos (una abuela y un abuelo), tres bisabuelos (dos bisabuelas y un bisabuelo) y así siguiendo…

Dado que Willy es muy perezoso él no quiere seguir haciendo más cálculos, por lo cual Maya decide ayudarlo construyendo el árbol genealógico de su gran amigo hasta el nivel cuatro.

Se anima a ayudar a Maya? Ud tiene que **construir** el árbol genealógico de Willy hasta el nivel n (el cuál es pasado como parámetro) e informar cuántas hembras y cuántos machos hay en su familia hasta ese nivel.

Ejemplo con n=4



Algoritmos y Estructuras de Datos – Curso 2011 (Redictado)
Parcial – 1era Fecha - Viernes 16 de Diciembre

Ejercicio 1.

Dada la siguiente recurrencia,

1.- Calcular el T(n) resolviendo la recurrencia, detallando los pasos seguidos para llegar al resultado.

2.- Calcular el O(n) justificando usando la definición de big-OH.

$$T(n) \begin{cases} 4 & n = 1 \\ 2T(n/2) + 5n + 1 & n > 1 \end{cases}$$

En el caso de ser necesario tenga presente las siguientes series:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}, \quad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}, \quad \sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4}$$

$$\sum_{i=1}^{n} i^4 = \frac{n(n+1)\left(6n^3+9n^2+n-1\right)}{30} \quad \sum_{i=0}^{n} c^i = \frac{c^{n+1}-1}{c-1}.$$

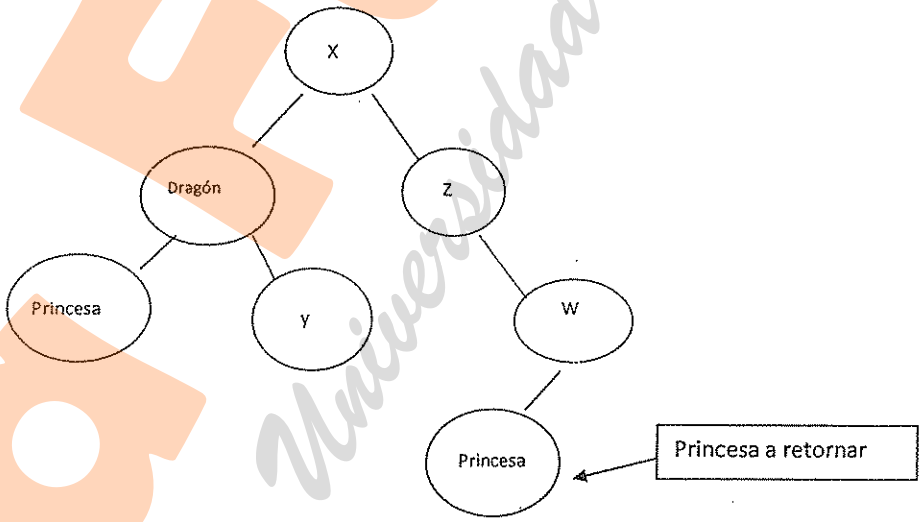
Ejercicio 2.

Supongamos disponible un tipo **Personaje** con las siguientes operaciones:

Personaje
- String tipo - String nombre
+esDragon (): boolean +esPrincesa (): boolean +setTipo (String unTipo)

Las operaciones **esDragon (): boolean** y **esPrincesa (): boolean** permiten averiguar si un personaje dado es un dragón o una princesa, respectivamente.
Suponemos que ningún personaje es dragón y princesa a la vez, y que un personaje puede no ser ninguna de las dos cosas.
Dado un árbol binario de personajes, se denominan nodos **accesibles** a aquellos nodos tales que a lo largo de la raíz hasta el nodo (ambos inclusive), NO se encuentra ningún dragón.
Debe implementar un método princesaAccesible():Personaje en la clase árbol binario que encuentre una princesa **accesible** lo más cerca posible de la raíz de un árbol dado,

Supongamos el siguiente ejemplo:



Ejercicio 3.

Una empresa de turismo realizó un mapa formado por un conjunto de ciudades conectadas por rutas. Para cada ciudad la empresa de turismo determinó la cantidad de días que los turistas deben pasar en la misma para visitar todos los sitios de interés. La empresa determina que se deben pasar esa cantidad de días, ni menos ni más, ya que menos implica desaprovechar la estancia porque no se visitan todos los lugares turísticos, y más días es perder el tiempo.
Considerando que la red de ciudades se representa por un grafo, en donde las ciudades son los nodos y las rutas las aristas que las conectan, debe implementar un método en la clase Grafo que reciba una cantidad total de días de vacaciones que desea tomarse y el algoritmo determine cuáles son las ciudades que debe visitar para aprovechar exactamente todos los días. Tenga presente que las ciudades deben ser contiguas (deben tener un camino que las conecta directamente), y que el circuito (camino) puede comenzar en cualquier ciudad. También considere que para una cantidad de días pueden existir distintos caminos posibles, con lo cual es suficiente con encontrar uno cualquiera.

Algoritmos y Estructuras de Datos - Curso 2011
1er Parcial - Sábado 11 de Junio

Ejercicio 1.
Dado el siguiente algoritmo,

a.- Calcular el T(n)

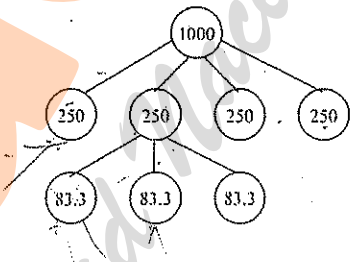
b.- Calcular el O(n) justificando debidamente.

```
for ( i=1; i<=n; i++) {
     for ( j=i^2; j>=1; j--) {
          for ( k=i; k<=j; k++)  {
               ** operacion de orden 1**;
               }
          }
     }
}
```

Pistas:

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4} \qquad \sum_{i=1}^{n} i^4 = \frac{n(n+1)(6n^3+9n^2+n-1)}{30}$$

Ejercicio 2.-
Sea una red de agua potable, la cual comienza en un caño maestro y el mismo se va dividiendo sucesivamente hasta llegar a cada una de las casas. Por el caño maestro ingresan 1000 litros y en la medida que el caño se divide, el caudal se divide en partes iguales en cada una de las divisiones. Es decir, si el caño maestro se divide en 4 partes, cada división tiene un caudal de 250 litros. Luego, si una de esas divisiones se vuelve a dividir en 3 partes, cada una tendrá un caudal de 83,3. La situación descripta se puede modelar de la siguiente forma a través del siguiente árbol general:



Usted debe implementar un método en la clase árbol general, que considerando que ingresan n litros por el caño maestro, calcule cual es el mínimo caudal que recibe una hoja.

Ejercicio 3.
Sea una red de empresas, en donde una empresa brinda servicios a muchas otras empresas sin ningún tipo de restricciones. Esta relación no es simétrica, es decir, la empresa A puede brindar servicios a la empresa B, sin embargo, B no necesariamente debe brindar servicios a la empresa A.

Usted es contratado por una nueva empresa que quiere ingresar a la Red. Por lo cual, le interesará determinar las 5 empresas de la red que llegan a la mayor cantidad de empresas. Una empresa A llega a otra empresa B, si le brinda servicios directa o indirectamente a través de otras empresas.

Algoritmos y Estructuras de Datos - Curso 2012
1er Parcial – Sábado 30 de Junio

Ejercicio 1.

Dado el siguiente fragmento de código:

```
public static int recu(int[] array, int count, int len) {
  if(len==0)
    return 0;
  else
    if (array[len-1]==count)
      return 1 + recu(array, count, len-1);
    else
      return recu(array, count, len-1);
}
public static void countOverlap(int[] arrayA, int[] arrayB) {
  int count = 0, calc = 0, tam=0
  if (arrayA.length == arrayB.length) {
    tam= arrayA.length;
    for (int i = 0; i < arrayA.length; i++)
      for (int j = 0; j < arrayB.length; j++)
        if (arrayA[i] == arrayB[j]) {
                count++;
                calc= calc + recu(arrayA,count,tam)+recu(arrayB,count,tam);
        }
  }
  System.out.println("count:"+count+"-calc:"+calc);
}
```

a.- Calcular el T(n) para el peor caso, detallando los pasos seguidos para llegar al resultado.

b.- Calcular el O(n) de la función del punto anterior justificando usando la definición de big-OH.

Ejercicio 2.

Una red binaria completa es una red que posee una topología de árbol binario completo (vea la Fig. 1 como ejemplo).

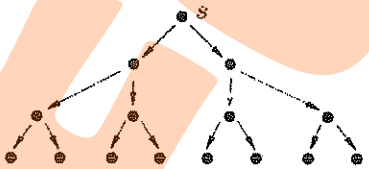


Figura 1. Ejemplo de una red binaria completa.

Los nodos que conforman una red binaria completa tienen la particularidad de que todos ellos conocen cual es su retardo de reenvío. El retardo de reenvío se define como el periodo comprendido entre que un nodo recibe un mensaje y lo reenvía a sus dos hijos.

Implemente un algoritmo que calcule el mayor retardo posible en el camino que realiza un mensaje desde la raíz hasta llegar a las hojas en una red binaria completa.

Nota: La red binaria completa no tiene siempre topología de árbol binario lleno.

Ejercicio 3.

Un laberinto es una estructura compleja formada por pasillos y encrucijadas que intenta confundir a quien en ella se adentra. Los laberintos son muy bien modelados mediante el uso de grafos conexos en los que la entrada y salida se diferencian del resto de los nodos. En la siguiente figura se muestra un ejemplo en donde la entrada (etiquetada con la letra E) y salida (con la letra S) se resaltan del resto de los nodos.

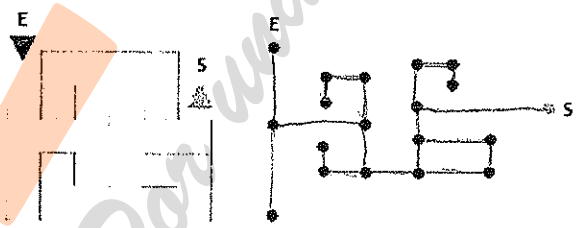


Figura 2.

Usted debe determinar cuál es el número mínimo de encrucijadas que debe atravesar una persona que se adentra en un laberinto en su camino a la salida. Considere que puede haber más de un camino que lo lleve a la salida.

Nota: La entrada y la salida son enviados como parámetros.

Ejercicio 1.

Dado un arreglo *array* de enteros con su longitud *len* mayor que cero y el siguiente método:

```
public static Boolean  sumatoria (int[] array, int len){

        If (len == 1) & (array [len-1] != 0)
             return false;
        else
           if (array [len] == suma (array, len%))
              return true;
          else  return sumatoria (array, len-1);
}

public static int suma (int[] array; int len){

        int count = 0;
        for (int i = 0, i <  len,  i++)
            count +=  array[i]
       return count;

}
```

a.- Calcular analíticamente el T(n) del método sumatoria, detallando los pasos seguidos para llegar al resultado.

b.- Calcular el O(n) de la función del inciso anterior usando la definición de big-OH

Ejercicio 2.

Estas a punto de hacer un trámite en oficinas de gobierno. Tenés que iniciar el trámite con una persona a la cuál llamamos persona 1.
Es de esperarse que esa persona no te resuelva el trámite directamente sino que te mande con otra, y a su vez esa con otra y así sucesivamente. Dependiendo del asunto, cada persona te puede mandar con otras, y tu trámite se realizará cuando alguna de ellas lo resuelva. Contas con un organigrama que representa este modelo jerárquico de atención gubernamental. La raíz de este organigrama es la persona 1, con la cual inicias tu trámite.
Los trabajadores de las oficinas no están conscientes de que puede haber trámites interminables, un trámite interminable es aquel en el cual vas a pedir informes con la persona 1, ella te manda con otra persona que a su vez te puede mandar con otra y así sucesivamente hasta que alguien te puede mandar con un empleado QueEncajonaTodo. Este prototipo de empleado es famoso en el ámbito estatal por no resolver nunca los trámites que llegan a ella!!
Te pedimos que escribas un método que encuentre el trámite interminable mas "largo", es decir, aquel en el que hables con la persona QueEncajonaTodo hablando antes con el mayor número de personas posibles. Devolviendo la secuencia de personas que forman el trámite y su longitud.

Ejercicio 3.

La Ciudad QuienSabeDónde ha sufrido un terrible terremoto, afortunadamente no hubo víctimas fatales, sin embargo, algunas de sus calles quedaron bloqueadas por escombros. Hay un grupo de rescatistas que se organizaron para ir a buscar a los accidentados que no pueden trasladarse y llevarlos al hospital de la ciudad para que les brinden atención médica. Los accidentados se han reunido en diferentes esquinas de la ciudad.
El grupo de rescatista cuenta con una única unidad móvil que tiene una determinada capacidad que usan para desplazarse por la ciudad. Esta unidad de rescate parte desde el hospital en busca de heridos, evitando las calles que quedaron obstruidas por los escombros. Los rescatistas siempre podrán llegar a cualquier esquina de la ciudad por algún camino, por lo cual los heridos no están preocupados por su rescate. Saben que pueden llegar a ellos.

La cantidad de accidentados es menor o igual a la capacidad de traslado de la unidad móvil.

El organizador de este proceso de asistencia lo considera finalizado cuando la unidad móvil completa su capacidad o se han rescatado a todas las víctimas, es decir cuando la unidad de rescate recorrió todas las esquinas de la ciudad quedando aún lugares disponibles en la unidad de traslado.

Debe escribir un método que recorra la ciudad y devuelva el camino con todas las esquinas por las que pasó la ambulancia en su recorrido por la ciudad además de las esquinas en las que subieron los heridos.

Considere que la ciudad está representada por un grafo conexo.

Algoritmos y Estructuras de Datos - Curso 2011
1er Parcial - Sábado 11 de Junio

Ejercicio 1.

Dado el siguiente algoritmo,

a.- Calcular el T(n)

b.- Calcular el O(n) justificando debidamente.

```
for ( i=1; i<=n; i++) {
      for ( j=i^2; j>=1; j--) {
            for ( k=i; k<=j; k++)  {
                  ** operacion de orden 1**;
                  }
            }
      }
}
```

Pistas:

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4} \qquad \sum_{i=1}^{n} i^4 = \frac{n(n+1)(6n^3+9n^2+n-1)}{30}$$

Solución:

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{i^2}\left(\sum_{k=i}^{j} c\right)\right) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i^2}(c(j-i+1))\right) = \sum_{i=1}^{n}\left(c\sum_{j=1}^{i^2}(j-i+1)\right) =$$

$$= \sum_{i=1}^{n}\left(c\left(\sum_{j=1}^{i^2} j - \sum_{j=1}^{i^2} i + \sum_{j=1}^{i^2} 1\right)\right) = \sum_{i=1}^{n}\left(c\left(\frac{i^2(i^2+1)}{2} - i^3 + i^2\right)\right) =$$

$$= c\sum_{i=1}^{n}\left(\frac{i^2(i^2+1)}{2} - i^3 + i^2\right) = c\sum_{i=1}^{n}\left(\frac{i^4}{2} + \frac{i^2}{2} - i^3 + i^2\right) =$$

$$= c\left(\frac{\frac{n(n+1)(6n^3+9n^2+n-1)}{30}}{2} + \frac{\frac{n(n+1)(2n+1)}{6}}{2} - \frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{6}\right) =$$

$$= c\left(\frac{n(n+1)(6n^3+9n^2+n-1)}{60} + \frac{n(n+1)(2n+1)}{12} - \frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{6}\right) =$$

$$= c\left(\frac{(n^2+n)(6n^3+9n^2+n-1)}{60} + \frac{(n^2+n)(2n+1)}{12} - \frac{n^2(n^2+2n+1)}{4} + \frac{(n^2+n)(2n+1)}{6}\right) =$$

$$= c\left(\frac{6n^5+9n^4+n^3-n^2+6n^4+9n^3+n^2-n}{60} + \frac{2n^3+n^2+2n^2+n}{12} - \frac{n^4+2n^3+n^2}{4} + \frac{2n^3+n^2+2n^2+n}{6}\right) =$$

$$= c\left(\begin{array}{l}\frac{6}{60}n^5+\frac{9}{60}n^4+\frac{1}{60}n^3-\frac{1}{60}n^2+\frac{6}{60}n^4+\frac{9}{60}n^3+\frac{1}{60}n^2-\frac{1}{60}n+\frac{2}{12}n^3+\frac{1}{12}n^2+\frac{2}{12}n^2+\frac{1}{12}n-\frac{1}{4}n^4-\frac{2}{4}n^3\\ -\frac{1}{4}n^2+\frac{2}{6}n^3+\frac{1}{6}n^2+\frac{2}{6}n^2+\frac{1}{6}n\end{array}\right)=$$

$$= c\left(\begin{array}{l}\frac{6}{60}n^5+\frac{9}{60}n^4+\frac{6}{60}n^4-\frac{1}{4}n^4+\frac{1}{60}n^3+\frac{9}{60}n^3+\frac{2}{12}n^3-\frac{2}{4}n^3+\frac{2}{6}n^3-\frac{1}{60}n^2+\frac{1}{60}n^2+\frac{1}{12}n^2+\frac{2}{12}n^2-\frac{1}{4}n^2\\ +\frac{1}{6}n^2+\frac{2}{6}n^2-\frac{1}{60}n+\frac{1}{12}n+\frac{1}{6}n\end{array}\right)=$$

$$= c\left(\begin{array}{l}\frac{6}{60}n^5+\frac{9}{60}n^4+\frac{6}{60}n^4-\frac{15}{60}n^4+\frac{1}{60}n^3+\frac{9}{60}n^3+\frac{10}{60}n^3-\frac{30}{60}n^3+\frac{20}{60}n^3-\frac{1}{60}n^2+\frac{1}{60}n^2+\frac{5}{60}n^2+\frac{10}{60}n^2-\frac{15}{60}n^2\\ +\frac{10}{60}n^2+\frac{20}{60}n^2-\frac{1}{60}n+\frac{5}{60}n+\frac{10}{60}n\end{array}\right)=$$

$$= c\left(\frac{6}{60}n^5+\frac{10}{60}n^3+\frac{20}{60}n^2+\frac{14}{60}n\right)=$$

$$\frac{6}{60}cn^5+\frac{10}{60}cn^3+\frac{20}{60}cn^2+\frac{14}{60}cn\_es\_O(n^5)\Rightarrow \frac{6}{60}cn^5+\frac{10}{60}cn^3+\frac{20}{60}cn^2-\frac{14}{60}cn\le kn^5\forall n\ge n_0\Rightarrow$$

$$\frac{6}{60}cn^5\le k_1n^5, k_1=\frac{6}{60}c, \forall n_0,$$

$$\frac{10}{60}cn^3\le k_2n^5, k_2=\frac{10}{60}c, \forall n_0$$

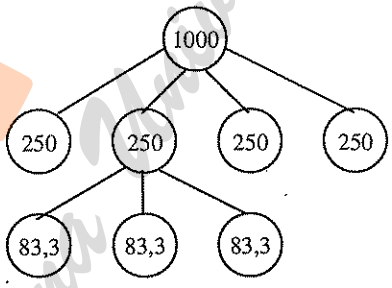
$$\frac{20}{60}cn^2\le k_3n^5, k_3=\frac{20}{60}c, \forall n_0$$

$$\frac{14}{60}cn\le k_4n, k_4=\frac{14}{60}c, \forall n_0$$

$$\frac{6}{60}cn^5+\frac{10}{60}cn^3+\frac{20}{60}cn^2+\frac{14}{60}cn\le kn^5, k=\left(\frac{6}{60}c+\frac{10}{60}c+\frac{20}{60}c+\frac{14}{60}c\right), \forall n_0$$

Ejercicio 2.-

Sea una red de agua potable, la cual comienza en un caño maestro y el mismo se va dividiendo sucesivamente hasta llegar a cada una de las casas. Por el caño maestro ingresan 1000 litros y en la medida que el caño se divide, el caudal se divide en partes iguales en cada una de las divisiones. Es decir, si el caño maestro se divide en 4 partes, cada división tiene un caudal de 250 litros. Luego, si una de esas divisiones se vuelve a dividir en 3 partes, cada una tendrá un caudal de 83,3. La situación descripta se puede modelar de la siguiente forma a través del siguiente árbol general:



Usted debe implementar un método en la clase árbol general, que considerando que ingresan n litros por el caño maestro, calcule cual es el mínimo caudal que recibe una hoja.

Solución:

```
double minimoCaudal (double caudalEntrante) {
      double answer = caudalEntrante;
      ListaGenerica<ArbolGeneral<T>> hijos = this.getHijos();
      if (!hijos.tamanio()==0) {
            hijos.comenzar()
```

```
            while (!hijos.fin()) {
                  answer = min(answer, hijos.elemento().minimoCaudal( caudalEntrante /
            hijos.tamanio());
                  hijos.proximo();
                  }
            }
      return (answer);
}
```

Ejercicio 3.

Sea una red de empresas, en donde una empresa brinda servicios a muchas otras empresas sin ningún tipo de restricciones. Esta relación no es simétrica, es decir, la empresa A puede brindar servicios a la empresa B, sin embargo, B no necesariamente debe brindar servicios a la empresa A.

Usted es contratado por una nueva empresa que quiere ingresar a la Red. Por lo cual, le interesará determinar las 5 empresas de la red que llegan a la mayor cantidad de empresas. Una empresa A llega a otra empresa B, si le brinda servicios directa o indirectamente a través de otras empresas.

Solución:

Debe hacer un DFS para cada una de las empresas de la red y contar los nodos a los que llega. Para ello, no debe olvidar marcar los nodos en la medida que visita, para evitar quedar en loop y contar varias veces el mismo nodo. Como el DFS se ejecuta una vez por cada nodo, es importante que antes de cada ejecución los nodos se vuelvan a marcar como no visitados.

Resolucion del Parcial: Curso de Verano 2011 Algor. y Estructura de Dat.

Ejercicio 2:

```
public void caminoOptimo (vertice<T> V) {
VERTICE [] visitados = new Vertice [n];
Vertice [] caminoPosible = new Vertice [n];
int dim = new int;
  Dfs ( V, visitados, caminoPosible);
  System.out.println ("El camino Optimo es:")
   + caminoPosible;
}
public void Dfs ( Vertice <T> V, Vertice [] visitados,
 Vertica [] caminoPosible) {
      V.setVisitado (true);
      if (V.getDATO = Destino) && (minimo(visitados))
        > (minimo (caminoPosible)) {
           caminoPosible = visitados;
        } else {
              Lista <arista> adyacentes =
                 V.getAdyacentes ();
              adyacents . comenzar ();
              while (! (adyacentes . Fin ())){
                    arista a = adyacentes.Proximo;
                  if (! a . getDestino(). esVisitado()))
                  { visitados [dim] = a.getDestino();
                    dim ++;
                    DFS (a.getDestino.(); } if
                } while
           }else
      dim --;
```

```
public int Minimo (Vertice [] auxiliar){
 int min = new int;
 min = 99;
 for (i = 0 ; i == auxiliar.length -1 ; i++){
          if (i < min) { min = i } } for
 Return i;
} Minimo.
```

Algoritmos y Estructuras de Datos - Curso 2011
1er Parcial - Sábado 11 de Junio

Ejercicio 1.

Dado el siguiente algoritmo.

a.- Calcular el T(n)

b.- Calcular el O(n) justificando debidamente.

```
for ( i=1; i<=n; i++) {
      for ( j=i^2; j>=1; j--) {
            for ( k=i; k<=j; k++)  {
                  ** operacion de orden 1**;
                  }
            }
      }
```

Pistas:

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \qquad \sum_{i=1}^{n} i^3 = \frac{n^2(n+1)^2}{4} \qquad \sum_{i=1}^{n} i^4 = \frac{n(n+1)(6n^3+9n^2+n-1)}{30}$$

Solución:

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{i^2}\left(\sum_{k=j}^{j} c\right)\right) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i^2}(c(j-i+1))\right) = \sum_{i=1}^{n}\left(c\sum_{j=1}^{i^2}(j-i+1)\right) =$$

$$= \sum_{i=1}^{n}\left(c\left(\sum_{j=1}^{i^2} j - \sum_{j=1}^{i^2} i + \sum_{j=1}^{i^2} 1\right)\right) = \sum_{i=1}^{n}\left(c\left(\frac{i^2(i^2+1)}{2} - i^3 + i^2\right)\right) =$$

$$= c\sum_{i=1}^{n}\left(\frac{i^2(i^2+1)}{2} - i^3 + i^2\right) = c\sum_{i=1}^{n}\left(\frac{i^4}{2} + \frac{i^2}{2} - i^3 + i^2\right) =$$

$$= c\left(\frac{\frac{n(n+1)(6n^3+9n^2+n-1)}{30}}{2} + \frac{\frac{n(n+1)(2n+1)}{6}}{2} - \frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{6}\right) =$$

$$= c\left(\frac{n(n+1)(6n^3+9n^2+n-1)}{60} + \frac{n(n+1)(2n+1)}{12} - \frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{6}\right) =$$

$$= c\left(\frac{(n^2+n)(6n^3+9n^2+n-1)}{60} + \frac{(n^2+n)(2n+1)}{12} - \frac{n^2(n^2+2n+1)}{4} + \frac{(n^2+n)(2n+1)}{6}\right) =$$

$$= c\left(\frac{6n^5+9n^4+n^3-n^2+6n^4+9n^3+n^2-n}{60} + \frac{2n^3+n^2+2n^2+n}{12} - \frac{n^4+2n^3+n^2}{4} + \frac{2n^3+n^2+2n^2+n}{6}\right) =$$

$$= c\left(\begin{array}{l}\frac{6}{60}n^5+\frac{9}{60}n^4+\frac{1}{60}n^3-\frac{1}{60}n^2+\frac{6}{60}n^4+\frac{9}{60}n^3+\frac{1}{60}n^2-\frac{1}{60}n+\frac{2}{12}n^3+\frac{1}{12}n^2+\frac{2}{12}n^2+\frac{1}{12}n-\frac{1}{4}n^4-\frac{2}{4}n^3\\ -\frac{1}{4}n^2+\frac{2}{6}n^3+\frac{1}{6}n^2+\frac{2}{6}n^2+\frac{1}{6}n\end{array}\right)=$$

$$= c\left(\begin{array}{l}\frac{6}{60}n^5+\frac{9}{60}n^4+\frac{6}{60}n^4-\frac{1}{4}n^4+\frac{1}{60}n^3+\frac{9}{60}n^3+\frac{2}{12}n^3-\frac{2}{4}n^3+\frac{2}{6}n^3-\frac{1}{60}n^2+\frac{1}{60}n^2+\frac{1}{12}n^2+\frac{2}{12}n^2-\frac{1}{4}n^2\\ +\frac{1}{6}n^2+\frac{2}{6}n^2-\frac{1}{60}n+\frac{1}{12}n+\frac{1}{6}n\end{array}\right)=$$

$$= c\left(\begin{array}{l}\frac{6}{60}n^5+\frac{9}{60}n^4+\frac{6}{60}n^4-\frac{15}{60}n^4+\frac{1}{60}n^3+\frac{9}{60}n^3+\frac{10}{60}n^3-\frac{30}{60}n^3+\frac{20}{60}n^3-\frac{1}{60}n^2+\frac{1}{60}n^2+\frac{5}{60}n^2+\frac{10}{60}n^2-\frac{15}{60}n^2\\ +\frac{10}{60}n^2+\frac{20}{60}n^2-\frac{1}{60}n+\frac{5}{60}n+\frac{10}{60}n\end{array}\right)=$$

$$= c\left(\frac{6}{60}n^5+\frac{10}{60}n^3+\frac{20}{60}n^2+\frac{14}{60}n\right)=$$

$$\frac{6}{60}cn^5+\frac{10}{60}cn^3+\frac{20}{60}cn^2+\frac{14}{60}cn\_es\_O(n^5)\Rightarrow \frac{6}{60}cn^5+\frac{10}{60}cn^3+\frac{20}{60}cn^2-\frac{14}{60}cn\le kn^5\forall n\ge n_0\Rightarrow$$

$$\frac{6}{60}cn^5\le k_1n^5, k_1=\frac{6}{60}c, \forall n_0,$$

$$\frac{10}{60}cn^3\le k_2n^5, k_2=\frac{10}{60}c, \forall n_0$$

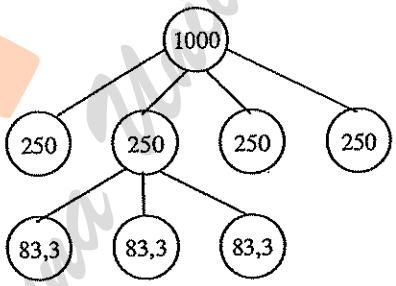
$$\frac{20}{60}cn^2\le k_3n^5, k_3=\frac{20}{60}c, \forall n_0$$

$$\frac{14}{60}cn\le k_4n, k_4=\frac{14}{60}c, \forall n_0$$

$$\frac{6}{60}cn^5+\frac{10}{60}cn^3+\frac{20}{60}cn^2+\frac{14}{60}cn\le kn^5, k=\left(\frac{6}{60}c+\frac{10}{60}c+\frac{20}{60}c+\frac{14}{60}c\right), \forall n_0$$

Ejercicio 2.-

Sea una red de agua potable, la cual comienza en un caño maestro y el mismo se va dividiendo sucesivamente hasta llegar a cada una de las casas. Por el caño maestro ingresan 1000 litros y en la medida que el caño se divide, el caudal se divide en partes iguales en cada una de las divisiones. Es decir, si el caño maestro se divide en 4 partes, cada división tiene un caudal de 250 litros. Luego, si una de esas divisiones se vuelve a dividir en 3 partes, cada una tendrá un caudal de 83,3. La situación descripta se puede modelar de la siguiente forma a través del siguiente árbol general:



Usted debe implementar un método en la clase árbol general, que considerando que ingresan n litros por el caño maestro, calcule cual es el mínimo caudal que recibe una hoja.

Solución:

```
double minimoCaudal (double caudalEntrante) {
	double answer = caudalEntrante;
	ListaGenerica<ArbolGeneral<T>> hijos = this.getHijos();
	if (!hijos.tamanio()==0) {
		hijos.comenzar()
```

UNLP. Facultad de Informática.
1/03/2011

Nombre y apellido: .....
Nro. Alumno: .....

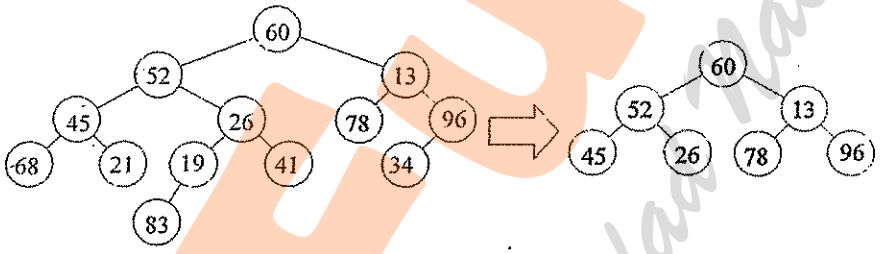
Algoritmos y Estructuras de Datos
Curso de Verano 2011
1era. Evaluación Práctica

Vértice<T>
- T dato - Lista<Arista<T>> adyacentes - int posicion
Vértice(T dato) setDato (T dato) getDato(): T setAdyacentes(Lista<Arista<T>> aristas) getAdyacentes(): Lista<Arista<T>> setPosicion(int pos) getPosicion(): int conectar(Vértice<T> vDest) conectar(Vértice<T> vDest, double peso)

Arista<T>
- Vértice<T> destino - double peso
Arista (Vértice<T> dest) setDestino(Vértice<T>vertice) getDestino():Vértice<T> setPeso(double peso) getPeso() :double

Ejercicio 3.

Especificar la operación **podar(): ArbolBinario<T>,** que toma como parámetro un árbol binario y devuelve otro árbol binario que es una copia de los niveles del árbol original hasta el nivel completo de mayor profundidad. Ej.:



ArbolBinario
- NodoBinario<T> raiz
+ArbolBinario() +ArbolBinario(T dato) -ArbolBinario(NodoBinario<T> nodo) -getRaiz():NodoBinario<T> +getDatoRaiz():T +getHijoIzquierdo():ArbolBinario<T> +getHijoDerecho():ArbolBinario<T> +agregarHijoIzquierdo(ArbolBinario<T> unHijo) +agregarHijoDerecho(ArbolBinario<T> unHijo) +eliminarHijoIzquierdo() +eliminarHijoDerecho() **+ podar(): ArbolBinario<T>**

NodoBinario
- T dato - NodoBinario<T> hijoIzquierdo - NodoBinario<T> hijoDerecho
NodoBinario(T dato) getDato():T getHijoIzquierdo():NodoBinario<T> getHijoDerecho():NodoBinario<T> setDato(T dato) setHijoIzquierdo(NodoBinario<T> unHijo) setHijoDerecho(NodoBinario<T> unHijo)

UNLP. Facultad de Informática.
1/03/2011

Nombre y apellido: .....
Nro. Alumno: .....

Algoritmos y Estructuras de Datos
Curso de Verano 2011
1era. Evaluación Práctica

Ejercicio 1.

Dado el siguiente método,

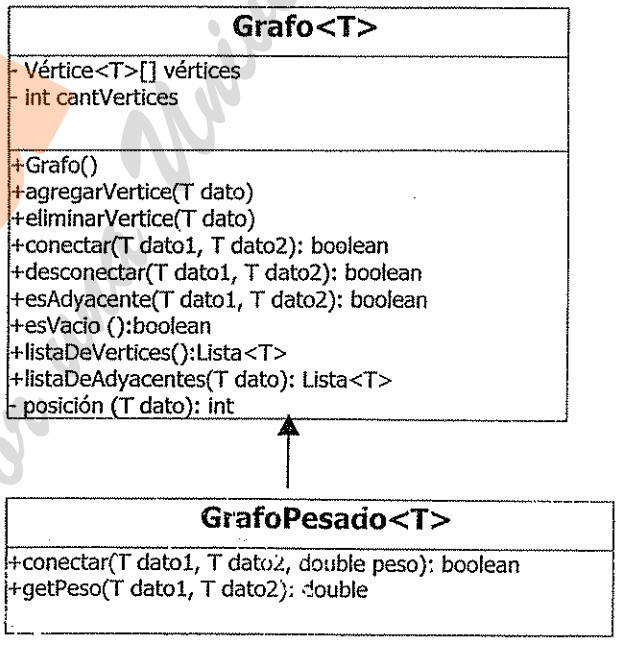
```
public class Parcial {

    public void  divVenc (int i, int j) {
          if (i = j)
              combina (i, j);
          else {
              m:= ( i+j ) / 2;
              a:= ( j – i ) / 4;
              divVenc (i, i+a);
              divVenc (i+a, m);
              divVenc (m, m+a);
              divVenc (m+a, j);
              combina (i, j);
          }
    }
    public void  combina (int i, int j) {
      for (int k = i; k < j; k++)
          Operacion();
    }
}
```

a) Plantear la función de recurrencia y calcular el T(n).
b) Calcular el O(n). Justificar el cálculo del mismo.

Ejercicio 2.

La ciudad de Cagitán va a instalar una planta de fusión nuclear, con paneles solares en los estacionamientos. Los componentes de la central son muy voluminosos y se requiere el uso de transportes especiales. Estos transportes se caracterizan porque necesitan caminos muy anchos, de varias decenas de metros. Por lo tanto, no nos importa la distancia del camino, sino que sea lo más ancho posible.
Buscamos el camino más ancho posible entre la fábrica de componentes y la ciudad de Cagitán. Tenemos un mapa de carreteras, con un conjunto de *n* puntos y *m* carreteras, donde las carreteras van entre dos puntos. Se supone que la fábrica está en el punto *1* y Cagitán en el *n*. El peso de cada arista *(i,j)* del grafo indica la anchura de la carretera de *i* a *j* (que es la misma que la que va de *j* a *i*).
Por ejemplo, si un camino pasa por cuatro carreteras de anchos: *50, 75, 18 y 24*, el ancho del camino es ***18***.
Escribir un algoritmo que resuelva el problema de manera óptima. Sugerencia: partiendo de algún algoritmo clásico de los vistos en clase, hacer las modificaciones necesarias para que resuelva este problema.



Algoritmos y Estructuras de Datos - Curso 2011
2do Parcial – Sábado 25 de Junio

Ejercicio 1.

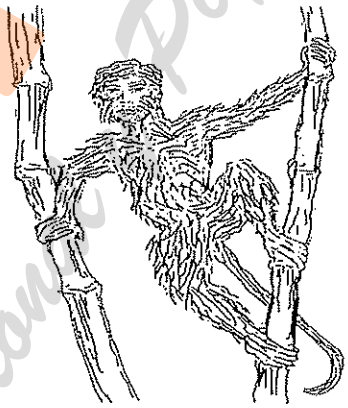
Dado el siguiente segmento de código,

a.- Calcular analíticamente el T(n), detalle los pasos seguidos para llegar al resultado.

b.- Calcular el O(n) justificando usando la definición de big-OH

```
j = 1;
while (j <= n) {
   for (i = n*n; i >=1; i = i-3)
          x=x+1;
j = j*2;
}
```

Ejercicio 2.



Había una vez un chimpancé llamado *Luchu Bandor,* cuyo significado era "Mono Playboy". *Luchu* estaba infelizmente casado con *Bunty Mona,* una chimpancé muy bonita pero de baja estatura. *Luchu* era alto y guapo, se sentía incómodo cuando estaba con *Bunty* en lugares públicos, ya que la gente los miraba a ellos continuamente. En un momento dado, *Luchu* no pudo soportar más esta situación y decidió hacer justicia a su nombre. Él comenzó a buscar una nueva esposa en el "Colegio Nacional de Señoritas Chimpancés". Cada día *Luchu* se subía a unas cañas de bambú y esperaba a que el ejercicio matutino empezara. Desde allí podía ver a todas las chimpancés haciendo su rutina de ejercicio diario. Ahora, *Luchu* estaba buscando a una chimpancé más alta pero que sea más baja que él, y también estaba interesado en aquella chimpancé un poco más alta que él. Sin embargo, alguien de su misma altura no la consideraba.

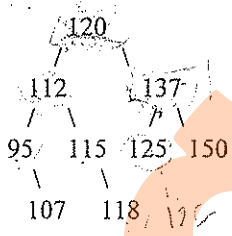
Luchu pudo modelar la situación descripta a través de un árbol AVL, el cuál contenía todas las alturas de las señoritas chimpancés que él había observado durante un cierto período de tiempo.

Su trabajo consiste en ayudar a *Luchu* para encontrar a las dos mejores chimpancés de acuerdo al criterio de selección establecido: la chimpancé más alta de las más bajas que él y la más baja entre las más altas que él.

Usted debe implementar un método en la clase árbol AVL, considerando que recibe como parámetro la altura de *Luchu* y debe devolver las alturas de las dos chimpancés buscadas ordenados de manera creciente. En el caso que sea imposible encontrar alguna de estas dos alturas devuelva un valor igual a 0 para la menor y 999 para la mayor. (0 y 999 no son alturas válidas, no están en el árbol)

Importante: considere que en el árbol existe una altura igual a la altura de *Luchu*.

Ejemplo

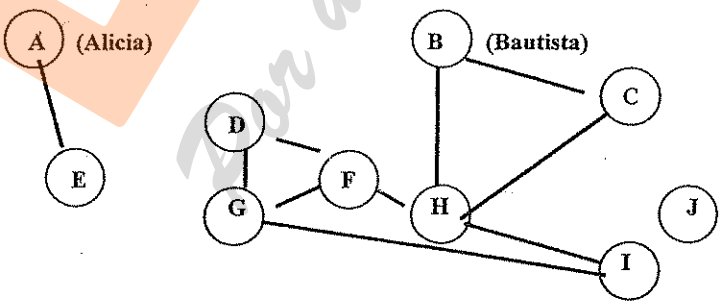


Las alturas del árbol están en cm.

- Si la altura de *Luchu* fuera 112, debe devolver : 107, 115
- Si la altura de *Luchu* fuera 120, debe devolver : 118, 125
- Si la altura de *Luchu* fuera 95, debe devolver : 0, 107
- Si la altura de *Luchu* fuera 150, debe devolver : 137, 999
- Si la altura de *Luchu* fuera 107, debe devolver : 95, 112
- Si la altura de *Luchu* fuera 118, debe devolver : 115, 120

Ejercicio 3.

Facebook, Friendster, etc son en la actualidad populares sitios web de redes sociales En estos sitios, una persona puede crear un perfil virtual y construir una relación de amistad con cualquier otra persona que esté registrada en la red, siempre y cuando ésta acepte su invitación. Esta relación de amistad es muy bien modelada usando un grafo, por ejemplo como el que se muestra en la figura siguiente (Por simplicidad los nombres de las personas fueron abreviados a un carácter).



En el ejemplo : Hay 3 grupos de amigos : {{BCDFGHI}, {AE} y {J} }

Algoritmos y Estructuras de Datos - Curso 2011 (Redictado)
Parcial – 1er Recuperatorio
Miércoles 8 de Febrero

jercicio 1.

da la siguiente recurrencia,

Calcular el T(n) resolviendo la recurrencia, detallando los pasos guidos para llegar al resultado.

Calcular el O(n) justificando usando la definición de big-OH.

$$T(n) \begin{cases} c & n = 0 \\ d & n = 1 \\ e + T(n-2) + n - 2 & n > 1 \end{cases}$$

jercicio 2.

ce tiempo que usted está dirigiendo su empresa y si bien le ha ido muy bien ya es tiempo de dejarle la tarea a un sucesor y que usted se dique a disfrutar de su familia y viajar. No es tarea fácil el buscar un sucesor. Para ello usted decide buscar al sucesor mas joven y que le pire su más profunda confianza. Es más, todos los descendientes que se encuentren entre usted y la persona que elija, también tienen que de su confianza.

nsidere que cuenta con la información de su árbol genealógico, donde usted es la raiz del árbol, y para cada descendiente posee la ormación de su edad. como así también sobre si es de confianza o no.

jercicio 3.

enso que algún día podré encontrar al Profesor Miguel quien me ha permitido organizar varios concursos, pero en realidad he fallado en las mis oportunidades. Lo último que sé es que él se encuentra en la mágica ciudad de la Esperanza. La ciudad de la Esperanza tiene uchas calles. Algunas son bidireccionales y otras son unidireccionales. Otra característica importante de estas calles es que algunas son ra personas menores de treinta años, y el resto son para los otros. Esto es para dar a los menores libertad en sus actividades. Cada calle ne una cierta longitud. Dada una descripción de tal ciudad y nuestras posiciones iniciales, debes encontrar el lugar más adecuado donde s podamos encontrar. El lugar más apropiado es el lugar en donde nuestros esfuerzos para llegar combinados sea el mínimo. Debes asumir e yo tengo 25 años y el Profesor Miguel más de 40.

DIJSTRA ↑

UNLP. Facultad de Informática.
10/03/2011

Nombre y apellido: .....
Nro. Alumno: .....

Algoritmos y Estructuras de Datos
Curso de Verano 2011
Recuperatorio de la Evaluación Práctica

Ejercicio 1.

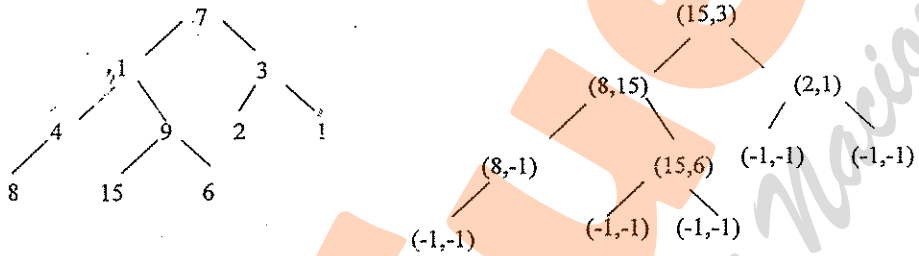
a) Resolver la siguiente recurrencia, dando el Orden del Tiempo de Ejecución obtenido :

$$T(n) = \begin{cases} 1 & \text{si } n = 0 \\ 8\,T(n-3) + 2^n & \text{si } n \geq 1 \end{cases}$$

b) Justificar el cálculo del Orden dado en el inciso a).

Ejercicio 2.

Implemente el método **anotarMaximos(): ArbolBinario<T>**, que dado un árbol binario de enteros positivos, devuelve un árbol binario en el que la información del nodo es un par de enteros que representa el mayor valor de cada subárbol. Si el subárbol es nulo, devuelve -1. P.ej., si el árbol de entrada es el de la izquierda, se devolverá el de la derecha:



ArbolBinario<T>
- NodoBinario<T> raiz
+ArbolBinario() +ArbolBinario(T dato) -ArbolBinario(NodoBinario<T> nodo) -getRaiz():NodoBinario<T> +getDatoRaiz():T +getHijoIzquierdo():ArbolBinario<T> +getHijoDerecho():ArbolBinario<T> +agregarHijoIzquierdo(ArbolBinario<T> unHijo) +agregarHijoDerecho(ArbolBinario<T> unHijo) +eliminarHijoIzquierdo() +eliminarHijoDerecho() **+ anotarMaximos(): ArbolBinario<T>**

NodoBinario<T>
- T dato - NodoBinario<T> hijoIzquierdo - NodoBinario<T> hijoDerecho
NodoBinario(T dato) getDato():T getHijoIzquierdo():NodoBinario<T> getHijoDerecho():NodoBinario<T> setDato(T dato) setHijoIzquierdo(NodoBinario<T> unHijo) setHijoDerecho(NodoBinario<T> unHijo)

Par
- int datoIzq - int datoDer
+ Par(int datoIzq, int datoDer) + setDatoDer(int datoDer) + setDatoIzq(int datoIzq) + getDatoDer(): int + getDatoIzq(): int

UNLP. Facultad de Informática.
10/03/2011

Nombre y apellido: .....
Nro. Alumno: .....

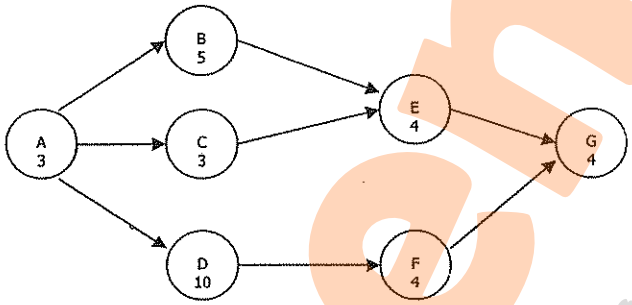
Algoritmos y Estructuras de Datos
Curso de Verano 2011
Recuperatorio de la Evaluación Práctica

Ejercicio 3.

Dado un grafo ***dirigido*** y ***acíclico*** que representa Plantas de generación de energía eléctrica ubicadas en ciudades (una planta por ciudad). Cada nodo indica la cantidad de energía que cada planta puede suministrar (en millones de KWh) y las aristas indican qué ciudades se le suministra la energía. Si de un nodo salen varias aristas, la energía se divide en partes iguales. Y si a un nodo le llegaran varias aristas, se suman todas las cantidades de las incidencias.

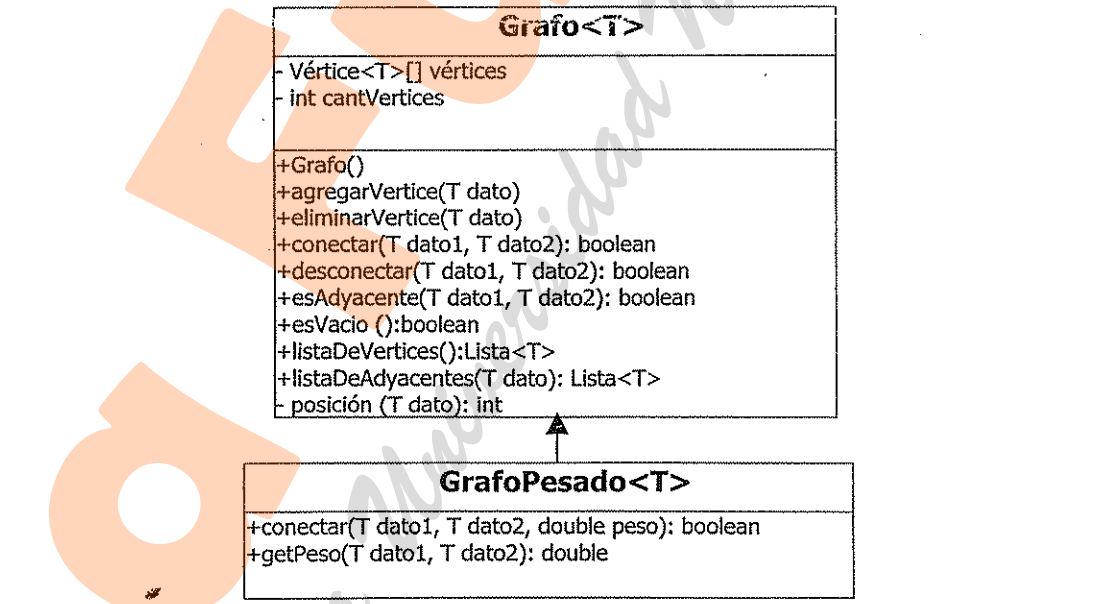
Implemente en la clase Grafo el método **suministroAlNodoFinal()**, que devuelve la cantidad de KWh que saldrían del nodo final (con adyacencias 0). Tenga en cuenta que puede haber más de un nodo final, en este caso debe devolver las cantidades de cada uno de ellos.

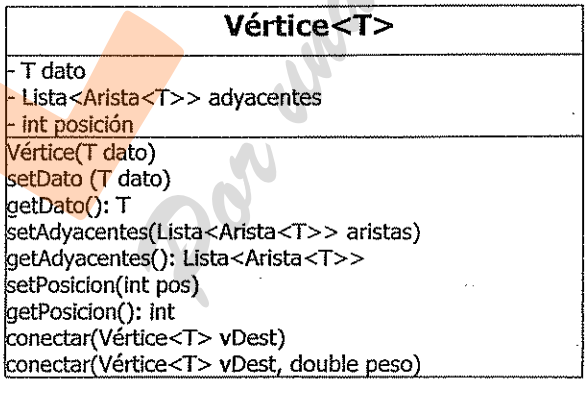
Ejemplo :

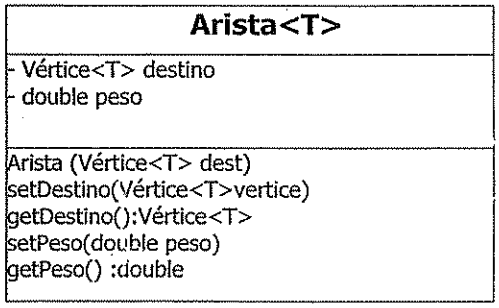


suministroAlNodoFinal () debería devolver 33.
Tenga presente que si al grafo anterior, le faltara el nodo G, tendría dos nodos finales y las energías serían: 14 para E y 15 para F.

Utilice las siguientes operaciones sin necesidad de implementarlas.







5.9.7 *The Stern-Brocot Number System*

PC/UVa IDs: 110507/10077, **Popularity:** C, **Success rate:** high **Level:** 1

The *Stern-Brocot tree* is a beautiful way for constructing the set of all non-negative fractions $\frac{m}{n}$ where m and n are relatively prime. The idea is to start with two fractions $(\frac{0}{1}, \frac{1}{0})$ and then repeat the following operation as many times as desired:

Insert $\frac{m+m'}{n+n'}$ between two adjacent fractions $\frac{m}{n}$ and $\frac{m'}{n'}$.

For example, the first step gives us one new entry between $\frac{0}{1}$ and $\frac{1}{0}$,

$$\frac{0}{1}, \frac{1}{1}, \frac{1}{0}$$

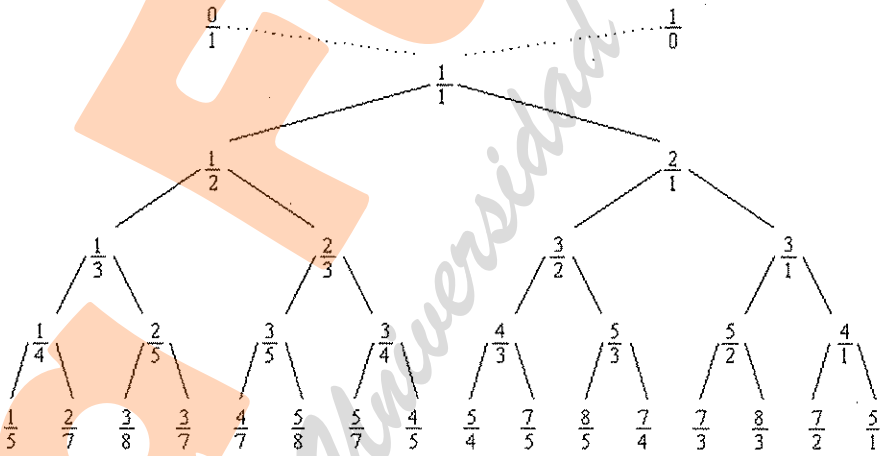
and the next gives two more:

$$\frac{0}{1}, \frac{1}{2}, \frac{1}{1}, \frac{2}{1}, \frac{1}{0}$$

The next gives four more:

$$\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}, \frac{3}{2}, \frac{2}{1}, \frac{3}{1}, \frac{1}{0}$$

The entire array can be regarded as an infinite binary tree structure whose top levels look like this–



This construction preserves order, and thus we cannot possibly get the same fraction in two different places.

We can, in fact, regard the *Stern-Brocot tree* as a *number system* for representing rational numbers, because each positive, reduced fraction occurs exactly once. Let us use the letters "L" and "R" to stand for going down the left or right branch as we proceed from the root of the tree to a particular fraction; then a string of L's and R's uniquely identifies a place in the tree. For example, LRRL means that we go left from $\frac{1}{1}$ down to $\frac{1}{2}$, then right to $\frac{2}{3}$, then right to $\frac{3}{4}$, then left to $\frac{5}{7}$. We can consider LRRL to be a representation of $\frac{5}{7}$. Every positive fraction gets represented in this way as a unique string of L's and R's.

Well, almost every fraction. The fraction $\frac{1}{1}$ corresponds to the empty string. We will denote it by I, since that looks something like 1 and stands for "identity."

In this problem, given a positive rational fraction, represent it in the *Stern-Brocot number system.*

Input

The input file contains multiple test cases. Each test case consists of a line containing two positive integers m and n, where m and n are relatively prime. The input terminates with a test case containing two 1's for m and n, and this case must not be processed.

Output

For each test case in the input file, output a line containing the representation of the given fraction in the *Stern-Brocot number system.*

Sample Input

```
5 7
878 323
1 1
```

Sample Output

```
LRRL
RRLRRLRLLLLRLRRR
```

FINAL DE ESTRUCTURAS DE DATOS – 12/03/2004

1) ¿Qué valor devuelve la siguiente función? Dar la respuesta como función en términos de n, y calcular su complejidad.

```
fun valor1(n : nat) dev r : nat
var i; j; k : nat
        r := 0 ;
        para i = 1 hasta n _ 1 hacer
                para j = i +_1 hasta n hacer
                        para k = 1 hasta j hacer
                                r := r + 1
                        fpara
                fpara
        fpara
ffun
```

2) Se tiene un algoritmo que tarda 2 segundos en resolver un problema de tamaño 1000. ¿Cuánto tardará en resolver un problema de tamaño 3000, si el orden del algoritmo es:

- $O(n^2)$
- $O(n \log n)$
- $O(n^3)$

3) Dado un árbol binario, la distancia entre dos nodos es la longitud del único camino que los conecta, y el diámetro del árbol es la distancia máxima sobre todos los pares de nodos. Desarrollar un algoritmo de tiempo lineal con respecto al número de nodos del árbol, para hallar el diámetro de un árbol binario dado.

4) a.- ¿Cuál es la cantidad mínima y máxima de elementos en una heap de altura h ?
b.- Implemente la operación BuildHeap, para construir una heap de n elementos en tiempo lineal.
c.- Explique la ejecución del algoritmo implementado con los siguientes datos:
1, 6, 9, 2, 7, 5, 2, 7, 4, 10

5) Se tiene un proyecto de construcción conformado por varias actividades: A, B, C, D, E, F y G con duraciones respectivas: 1, 2, 5, 3, 4, 3, 2. Algunas actividades preceden a otras en su ejecución, como se indica a continuación:

Actividad	Precede a	Actividad
A		B, C
B		D, E, F
C		E
D		F

Actividad	Precede a	Actividad
E		F, G
F		G
G		-

Determine los tiempos más tempranos en que debe comenzar cada actividad para que el proyecto se ejecute en el menor tiempo posible.

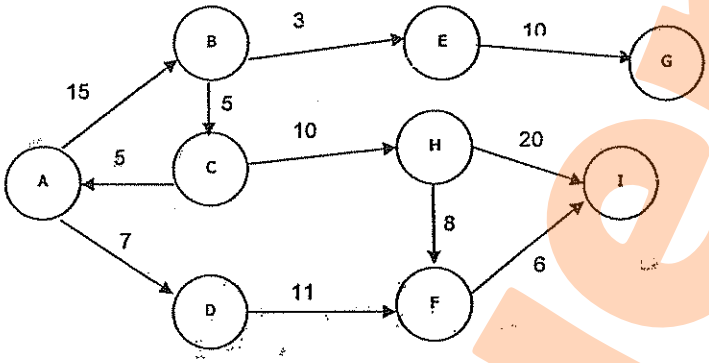
UNLP. Facultad de Informática. Nombre y apellido: .....
17/03/2011 Nro. Alumno: .....

Algoritmos y Estructuras de Datos
Curso de Verano 2011

Recuperatorio del TEMA GRAFOS

Dado un grafo orientado y pesado positivamente, y ***dos*** nodos de dicho grafo, implementar un método que encuentre y muestre (si existe) un camino simple de exactamente costo total c ($c > 0$).
Recordar que un *camino simple* es aquél en el que todos los nodos son distintos (excepto el primero y el último que pueden ser iguales)

Ejemplos :
1.- $C = 50$ y los nodos ***A*** e ***I*** --- El camino simple de costo 40 es : A – B – C – H – I
2.- $C = 25$ y los nodos ***A*** e ***I*** --- No existe el camino simple de costo 25 entre los nodos A e I
3.- $C = 28$ y los nodos ***A*** y ***E*** --- No existe el camino simple de costo 28 entre los nodos A y E



Utilice las siguientes operaciones sin necesidad de implementarlas.

Grafo<T>

- Vértice<T>[] vértices
- int cantVertices

+Grafo()
+agregarVertice(T dato)
+eliminarVertice(T dato)
+conectar(T dato1, T dato2): boolean
+desconectar(T dato1, T dato2): boolean
+esAdyacente(T dato1, T dato2): boolean
+esVacio ():boolean
+listaDeVertices():Lista<T>
+listaDeAdyacentes(T dato): Lista<T>
- posición (T dato): int

GrafoPesado<T>

+conectar(T dato1, T dato2, double peso): boolean
+getPeso(T dato1, T dato2): double

Vértice<T>

- T dato
- Lista<Arista<T>> adyacentes
- int posición

Vértice(T dato)
setDato (T dato)
getDato(): T
setAdyacentes(Lista<Arista<T>> aristas)
getAdyacentes(): Lista<Arista<T>>
setPosicion(int pos)
getPosicion(): int
conectar(Vértice<T> vDest)
conectar(Vértice<T> vDest, double peso)

Arista<T>

- Vértice<T> destino
- double peso

Arista (Vértice<T> dest)
setDestino(Vértice<T>vertice)
getDestino():Vértice<T>
setPeso(double peso)
getPeso() :double

UNLP. Facultad de Informática.

Algoritmos y Estructuras de Datos 13 de Febrero
APREF 2013

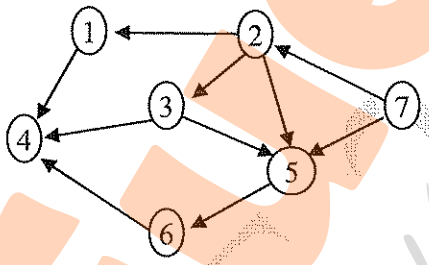
Encuentro 3: Grafos
Ordenación topológica – Recorridos – Algoritmos de Caminos mínimos (grafos no pesados)

Ejercicio 1

(a) ¿Qué es la ordenación topológica de un grafo dirigido acíclico?
(b) ¿Cuáles son las estructuras de datos posibles para almacenar los datos durante el proceso de ordenación? ¿Cómo influye cada una de ellas en el cálculo del tiempo de ejecución?

Ejercicio 2

Mostrar una ordenación topológica para el siguiente grafo dirigido acíclico. ¿La ordenación obtenida es única? En caso negativo mostrar otra ordenación válida.



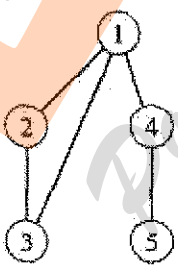
Ejercicio 3

Se tiene un proyecto formado por varias tareas: A, B, C, D, E, F, y G con duraciones respectivas 2, 3, 6, 4, 5, 4 ,3. Algunas tareas preceden a otras en su ejecución, como se indica a continuación: A precede a B y C, B precede a D y E , C precede a E y G, D precede a E, E precede a F y G, F precede a G.
Determine los tiempos más tempranos en que debe comenzar cada tarea, para que el proyecto se ejecute en el menor tiempo posible.

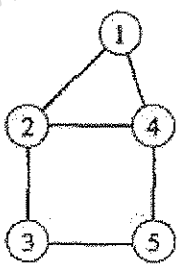
Ejercicio 4

Para cada uno de los siguientes grafos no dirigidos, determine si los nodos pueden ser visitados en el orden 1, 2, 3, 4, 5 cuando el grafo es recorrido usando: DFS, BFS, ambos o ninguno. Justifique brevemente su respuesta en cada caso.

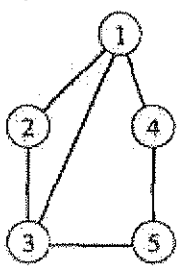
A)



B)



C)



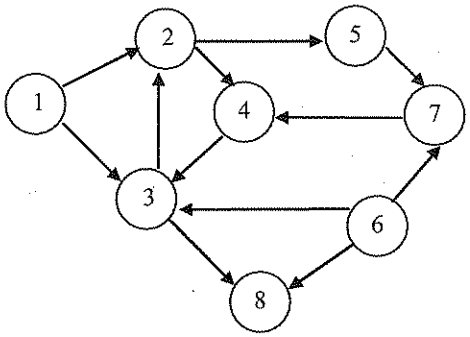
UNLP. Facultad de Informática.

Algoritmos y Estructuras de Datos 13 de Febrero
APREF 2013

Encuentro 3: Grafos
Ordenación topológica – Recorridos – Algoritmos de Caminos mínimos (grafos no pesados)

Ejercicio 5

Dado el siguiente grafo dirigido:



a.- Dibuje el árbol abarcador del DFS indicando los distintos tipos arcos que aparecen.
b.- Describa una estrategia para reconocer si un arco es "*arco de avance*" ó "*arco forward*".
c.- ¿Es fuertemente conexo? Justifique su respuesta

Ejercicio 6

Sea G un grafo dirigido no pesado y sea x un vértice de G; se define ***excentricidad*** de x en G, y se denota por Exc(x), como:

Exc(x) = Maximo(distancia_mínima(x,y)), $\forall y \in G$

Implemente un algoritmo para calcular la ***excentricidad*** de un vértice.

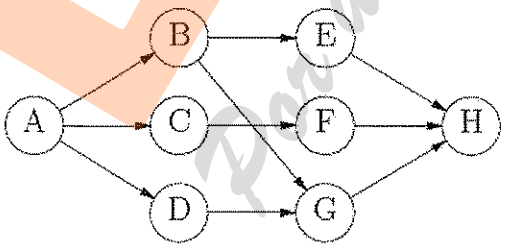
Ejercicio 7

Implemente un algoritmo para calcular la ordenación topológica de un DAG, utilizando el recorrido DFS.
Pista: La numeración se hace en orden inverso comenzando desde |V|
(|V| es el total de vértices del grafo)

Ejercicio 8

Una solución clásica para determinar las componentes fuertemente conexas de un grafo dirigido G es usar dos recorridos en profundidad (DFS). El primero determina el orden en que los nodos serán considerados en el siguiente paso. El segundo recorrido trabaja con el grafo transpuesto y construye un bosque donde cada árbol representa una componente conexa. El grafo transpuesto G^T puede construirse invirtiendo todas las aristas del grafo original.

a.-Indique la numeración de los vértices que surge a partir del recorrido DFS (sobre el dibujo del grafo)
b.-Identifique las componentes fuertemente conexas del siguiente grafo.



UNLP. Facultad de Informática.

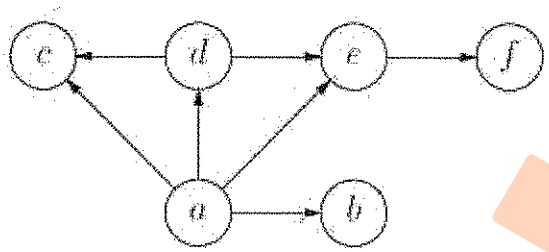
Algoritmos y Estructuras de Datos 13 de Febrero
APREF 2013

Encuentro 3: Grafos
Ordenación topológica – Recorridos – Algoritmos de Caminos mínimos (grafos no pesados)

Multiple Chioce

1.- ¿Cuál de la/s siguiente/s opción/es **no** es un posible orden para un recorrido DFS del grafo de la figura?

(a) abcdef (b) abdefc (c) adcefb (d) adefbc (e) aefdcb



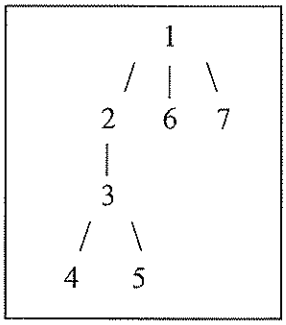
2.- Sea un grafo no dirigido con n vértices. Entonces:

a) Si se hace un recorrido en amplitud partiendo del nodo x, el conjunto de vértices alcanzados es igual al conjunto total de vértices del grafo.
b) Si se hace un recorrido en amplitud partiendo del nodo x, el conjunto de vértices alcanzados al final será igual al que resultaría si el recorrido fuera en profundidad.
c) Si se hace un recorrido en amplitud partiendo del nodo x, no puede asegurarse que el conjunto de vértices alcanzados al final resulte el mismo que si el recorrido fuera en profundidad.
d) Si se hace un recorrido en amplitud partiendo del nodo x y el grafo es cíclico se entra en un bucle infinito.
e) Si se hace un recorrido en amplitud partiendo del nodo x y el conjunto de vértices alcanzados resultante no es igual al conjunto total de vértices del grafo es porque el grafo no es conexo.

3.- El recorrido en profundidad de un grafo G no dirigido ha producido el árbol que se muestra en la figura, en el que cada nodo está numerado siguiendo el orden de visita del recorrido en profundidad.

Cuál de las siguientes afirmaciones es correcta?

(a) El nodo 6 es adyacente al nodo 4.
(b) El nodo 2 puede ser adyacente al nodo 5, y el nodo 4 puede ser adyacente al nodo 1.
(c) El nodo 6 y 7 no son adyacentes, y el nodo 5 y el nodo 7 si lo son.
(d) El nodo 1 sólo puede ser adyacente a los nodos 2, 6 y 7.
(e) Se trata de un grafo fuertemente conexo.

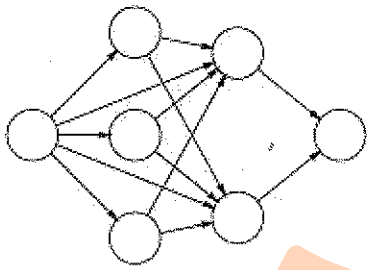


Encuentro 3: Grafos
Ordenación topológica – Recorridos – Algoritmos de Caminos mínimos (grafos no pesados)

4.- ¿Bajo qué condición podemos concluir que un grafo dirigido no tiene orden topológico?

(a) Cada vértice tiene al menos una arista saliente
(b) Cada vértice tiene al menos una arista entrante
(c) El grafo tiene un ciclo
(d) Cualquiera de las condiciones mencionadas implica que el grafo no tiene orden topológico

5.- ¿Cuántos órdenes topológicos distintos tiene el siguiente grafo?



(a) 5 (b) 6 (c) 9 (d) 12 (e) 15

UNLP. Facultad de Informática.

Algoritmos y Estructuras de Datos 8 de Febrero

APREF 2013

Encuentro 2: Árboles

Ejercicio 1

Defina árbol binario completo y árbol binario lleno. ¿Es verdad que todo árbol binario completo es lleno? ¿Y viceversa?

Ejercicio 2

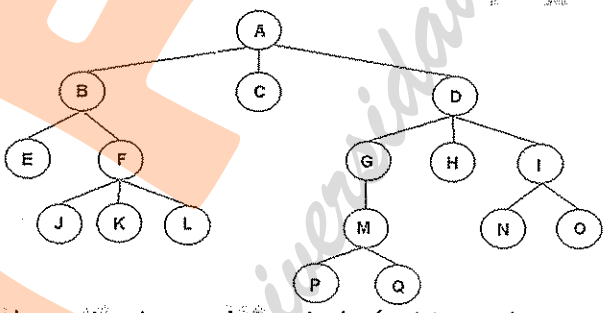
Defina la función de tiempo de ejecución del recorrido preorden en un árbol binario. Calcule el O(n) de la función definida.

Ejercicio 3

a.- Dada la siguiente expresión postfija : **I J K + + A B * C - *** , dibuje su correspondiente árbol binario de expresión
b.- Convierta la expresión **((a + b) + c * (d + e) + f) * (g + h)** en expresión prefija

Ejercicio 4

La siguiente figura muestra un árbol general:



1) Complete los blancos de las sentencias con la terminología vista en clase.

   a) ..... es la raíz del árbol.
   b) ..... es padre de B, C y D.
   c) .....y ..... son hermanos, puesto que ambos son hijos de B.
   d) ....y ..... son las hojas del árbol.
   e) El camino desde A a J es único, lo conforman los nodos .... y es de largo .....
   f) .....es ancestro de P, y por lo tanto ..... es descendiente de D.
   g) L no es descendiente de C, puesto que no existe ....desde C a L.
   h) La profundidad/nivel de C es ....., de F es ..... y de ..... es 4.
   i) La altura de C es ....., de ..... es 1 y de D es .....
   j) La altura del árbol es 4 (largo del camino entre la ..... y .....).

2) Muestre gráficamente las siguientes *representaciones*, aplicadas al árbol de la figura :
   a) Lista de hijos
   b) Hijo más izquierdo-hermano derecho

3) Aplique los recorridos :
   a) en profundidad
      i) preorden

iii) postorden
b) por niveles

Ejercicio 5

Defina la función de tiempo de ejecución del recorrido preorden en un árbol general. Exprese la función de dos formas:

- teniendo en cuenta los nodos
- teniendo en cuenta la altura

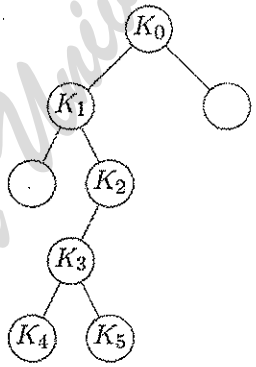
Ejercicio 6

Muestre el árbol binario de búsqueda que resulta de insertar las claves siguientes en un árbol inicialmente vacío

50, 40, 45, 47, 46, 41, 35

Ejercicio 7

En el siguiente árbol binario de búsqueda, sólo aparecen algunos valores en los nodos en forma simbólica. Suponga que se elimina k_0, y que el método que se usa involucra al predecesor inorden. ¿Qué clave quedará en la raíz en el árbol resultante?



Ejercicio 8

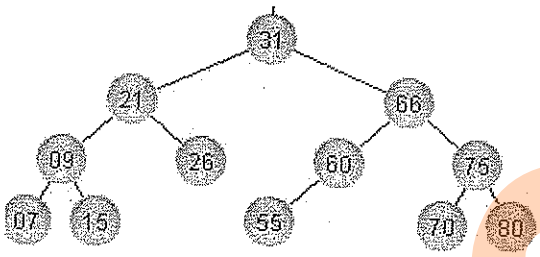
En el siguiente árbol binario de búsqueda, se deben realizar 2 comparaciones antes de encontrar el valor "26". Considerando el mismo conjunto de datos, determine la distribución de los mismos en dos árboles binarios de búsqueda de manera que el valor "26" tome:
i) la **mejor ubicación** y
ii) la **peor ubicación**.

UNLP. Facultad de Informática.

Algoritmos y Estructuras de Datos 8 de Febrero

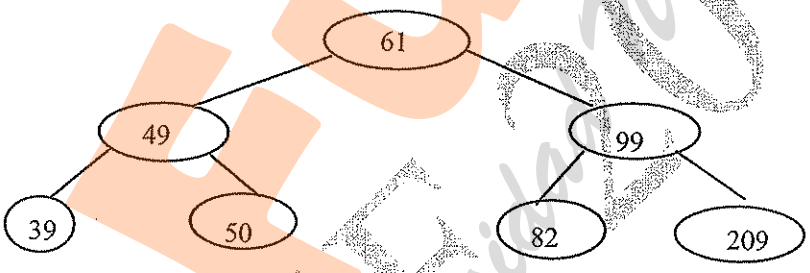
APREF 2013

Encuentro 2: Árboles



Ejercicio 9

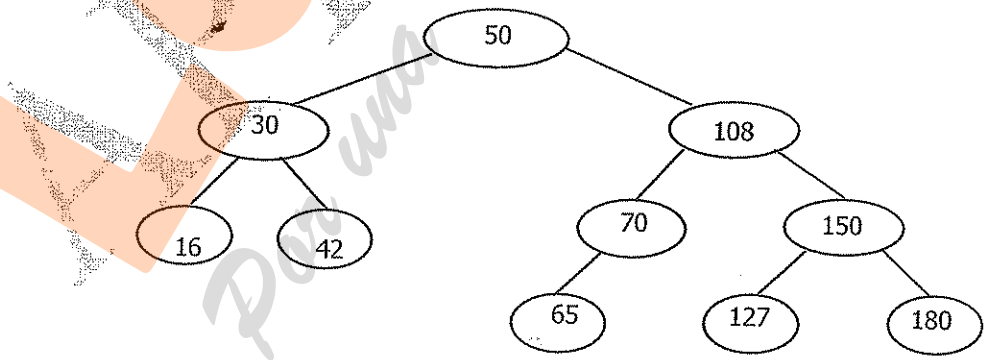
a) **Modificar** el siguiente AVL, insertando (I) las siguientes claves: I(22),I(101),I(17),I(46),I(3),I(24),I(28). Indicar en los casos en que sea necesario que tipo de rotación se realizó.



b) **Modificar** el AVL anterior, insertando (I) y borrando (B) las siguientes claves: I(22),I(101),B(50), I(17),I(46),I(3),I(20),I(25),B(82),B(39). Indicar en los casos en que sea necesario que tipo de rotación se realizó.

Ejercicio 10

Dado el siguiente árbol AVL:



a.- ¿Qué clave debería insertar en el siguiente árbol AVL para que se produzca un desbalanceo que se resuelve con una ***Rotación Simple a Derecha***

Encuentro 2: Árboles

c.- ¿De qué nodo/s podría ser hija la clave que produzca el desbalanceo mencionado en el inciso a.-?
d.- Inserte en el árbol una de las claves y dibuje las modificaciones que la rotación provoca.

Ejercicio 11

¿Cuántas rotaciones requieren las operaciones de inserción y borrado en un árbol AVL?

Ejercicio 12

Se tiene una Min-Heap con los siguientes elementos: 10, 30, 20, 40, 50, 70, 80, 60. Si se inserta el 25, ¿en qué posición quedará después de completada la inserción?

Ejercicio 13

En un sistema operativo multitarea se encuentran ejecutándose múltiples tareas organizadas en una cola de prioridades. En un determinado momento se desea bajar la prioridad de una tarea que consume muchos recursos. Implemente el algoritmo asumiendo que conoce la ubicación de la tarea dentro de la cola de prioridades.

Ejercicio 14

a.- Mostrar la ejecución de la operación CrearHeap(a,n) para cada una de las siguientes implementaciones, tomando como entrada el arreglo a={1,6,9,2,7,5,2,7,4,10}

```
public void CrearHeap1 (int[] a, int n) {
       int  i;
       for (i=2; i ≤ n; i++)
            filtradoArriba(a,n,i)

}
```

```
public void CrearHeap2 (int[] a, int n){
       int i;
       for (i=(n div 2); i≥1; i--)
           FiltradoAbajo(A,n,i)

}
```

b.- Indique cómo calcularía el T(n) en cada caso.
c.- ¿Qué implementación considera más eficiente? Justifique su elección.

Multiple choice

1.- Suponga que para un árbol binario T, el último nodo en postorden es el mismo que el último nodo en inorden, ¿qué se puede concluir?

(a) El subárbol izquierdo de T es vacío
(b) El subárbol derecho de T es vacío
(c) Ningún nodo en el árbol tiene dos hijos
(d) Hay a lo sumo 3 nodos en el árbol

UNLP. Facultad de Informática.

Algoritmos y Estructuras de Datos 8 de Febrero

APREF 2013

Encuentro 2: Árboles

2.- Dado el siguiente recorrido en postorden de un árbol de expresión

D A E * * B C * +

¿Cuáles dos letras están más alejadas de la raíz del árbol?

(a) A y B (b) B y C (c) C y D (d) D y E (e) Otras

3.- Suponga que un nodo ***x*** en un árbol AVL tiene altura 8. ¿Cuál de las siguientes sentencias describen la situación de la altura del padre de ***x***?

(a) Es 7 (b) Es 8 (c) Es 9 (d) Es 10 (e) Puede ser 9 ó 10

4.- ¿Cuál sería la max-heap resultante después de haber insertado los siguientes elementos en el orden indicado: 5,3,8,2,1.

(a)

1	2	3	4	5	6
8	2	3	5	1	...

(b)

1	2	3	4	5	6
8	3	1	5	2	...

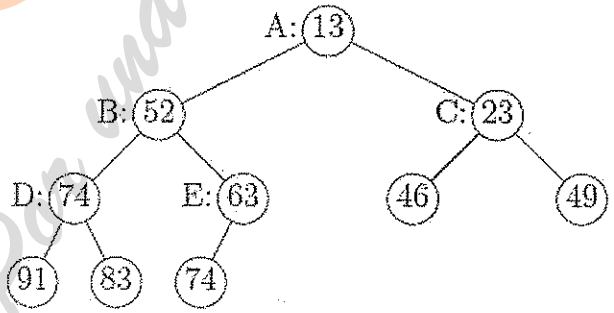
(c)

1	2	3	4	5	6
8	3	5	2	1	...

(d)

1	2	3	4	5	6
8	2	5	1	3	...

5.- Dada la siguiente cola de prioridades almacenada en una heap. Si se inserta la clave 31, ¿en qué posición de la heap quedará?



(a) A (b) B (c) C (d) D (e) E