



Practicas General

Programación (Universidad Mayor de San Simón)



Escanea para abrir en Studocu

The background of the page features a large, faint watermark of the Universidad Mayor de San Simón (UMSS) logo. It is an inverted triangle with a red border and two red triangles at the top corners. The letters 'UMSS' are written in a bold, sans-serif font across the top. In the center of the triangle is a smaller emblem.

UMSS

Guia de Prácticas

Elementos de Programación y Estructuras de Datos

Universidad Mayor de San Simón

Departamento de Informática - Sistemas

Ma. Leticia Blanco Coca

2018

Índice general

1. Programación Orientada a Objetos	2
2. Recursividad	7
3. Estructuras de Datos Lineales	45
4. Estructuras de Datos No Lineales	60

Capítulo 1

Programación Orientada a Objetos

Dadas las siguientes descripciones, define cuales son candidatas a clase y cuales candidatas a objetos, por qué? Define el modelo que mostraría el contexto y sus relaciones.

1. En una villa se tienen varios pobladores que viven en armonía, ya que suelen compartir el lugar de trabajo, y los lugares de distracción. Juan Pablo es uno de los habitantes de esta villa que suele ir mucho al Parque “El aguilita” ya que tiene muchos juegos infantiles.
2. Una empresa de telecomunicaciones se caracteriza por brindar diferentes servicios, tales como la telefonía celular. Una de sus unidades se encarga de atender a los clientes que tiene la empresa y ayuda a resolver problemas que los mismos tengan con algún servicio que ofrecen. La velocidad de respuesta de los operadores de atención al cliente es crucial ya que esto puede generar una llamada de atención por la superintendencia de telecomunicaciones. Se dio un caso en el que Maria (una operadora) se tardó en atender una solicitud de reclamo mas de 20 días por lo cual Honorato Orfebre, uno de los mejores clientes elevó una queja.
3. Un sistema de identificación de personas en la oficina de lucha contra el crimen desea tener formas más rápidas de construir un dibujo que muestre los rasgos de un sospechoso. Se sabe que los dibujos de las caras se dividen en tres regiones: la parte alta, la parte media y la baja. En la parte media se encuentran elementos como el cabello, la frente, los ojos, las cejas. En la parte media se describe la nariz y en la parte baja la boca y barbilla. De cada elemento se puede tener a su vez ciertos modelos establecidos como: boca grande y carnosa o boca chica y carnosa.

En esta sección se plantean descripciones de dominio los mismos que deben ser resueltos utilizando POO.

4. Transfer es una empresa de transporte público, que tiene una flota de taxis y una variedad de líneas de buses. Los taxis tienen un nombre que los identifica (por ejemplo: Rojo 100), una ubicación actual, el costo de pasaje básico, el costo adicional por persona, el estado (ocupado/libre) y la recaudación del día. De los buses se conoce el nombre de la línea, la ruta, el costo del pasaje básico y la recaudación del día. Los buses debido a que se tiene un costo único se recoge pasajeros de uno en uno, pero los taxis pueden recoger a uno o a muchos pasajeros. Los vehículos deberían poder indicar si pueden llevar a un pasajero de X a Y. La ruta se considera como una secuencia ordenada de lugares que se puede visitar. La empresa desea automatizar algunos de sus procesos, mismos que se detallan a continuación:

- Se requiere saber las recaudaciones del día de la empresa
- Se quiere saber la lista de vehículos que pueden llevar de un lugar X a Y
- Se desea automatizar el recojo de pasajeros

5. Se desea tener información de los ambientes que se destinan para dar clases en la FCYT. Nos interesa la capacidad de los ambientes y la ubicación vertical en la que se encuentran. La ubicación vertical hace referencia al piso en el que se encuentra un ambiente.

Se requiere saber:

- Cuántos ambientes existen con capacidad mayor a X y que estén en la ubicación U
- Cuántos ambientes en la ubicación X existen
- Emitir un reporte de todos los ambientes que tiene una facultad
- Emitir un reporte de los primeros n ambientes que tengan una capacidad mínima de X
- Indicar si es posible a nadir un ambiente más a una facultad
- Permitir ingresar un nuevo ambiente mas
- Realizar un reporte que indica la suma de todas las capacidades por ubicación

Se sabe que las Facultades tienen a lo sumo 50 ambientes y el edificio mas alto tiene 10 pisos. Realiza el modelo y los métodos que permitan satisfacer los requerimientos

Dados los siguientes códigos y modelos se pide encontrar una descripción de lo que resuelven y/o hacen además de completar los métodos que se indican.

6. Dado el siguiente código, indica a qué contexto pertenece, reconstruye el modelo y completa los cuerpos de los métodos que faltan:

```
import java.util.ArrayList;
/**
 * clase que identifica una coleccion de
 * elementos comparables
 * @author MLBC
 * @version 5.09.07
 */
public class Coleccion
{
    private ArrayList<Comparable> datos;
    public Coleccion()
    {
        datos = new ArrayList<Comparable>();
    }
    public int tamano()
    {
        return datos.size();
    }
    public Comparable dato(int pos)
    {
        return datos.get(pos);
    }
    public void agregar(Comparable dato)
    {
        datos.add(dato);
    }
    public Comparable buscar(Buscador buscador, Comparable dato)
```

```
{
    return buscador.buscar(this, dato);
}
}
/**
 * clase que identifica objetos
 * que saben buscar un dato en una
 * coleccion
 * @author MLBC
 * @version 05.09.07
 */
abstract public class Buscador
{
    /**
     * Metodo que busca un dato en la coleccion
     * @param coleccion la coleccion donde se buscara
     * @param dato el dato que se buscara
     *
     * @return el dato si se ha encontrado
     */
    abstract public Comparable buscar(Coleccion coleccion, Comparable dato);
}
/**
 * clase que identifica a buscadores secuenciales
 * @author MLBC
 * @version 05.09.07
 */
public class Secuencial extends Buscador
{
    public Secuencial()
    {}
    public Comparable buscar(Coleccion coleccion, Comparable dato)
    {
        Comparable res = null;
        int cantidad = coleccion.tamano();
        int posicion = 0;
        boolean encontrado = false;
        while(posicion < cantidad && !encontrado)
        {
            if(coleccion.dato(posicion).compareTo(dato)==0)
            {
                res = coleccion.dato(posicion);
                encontrado = !encontrado;
            }
            posicion++;
        }
        return res;
    }
}
```

```
/**
 * Escribe la clase secuencial acotado
 * @author
 * @version
 */
public class SecuencialAcotado extends Buscador
{
    public SecuencialAcotado()
    {}
    public Comparable buscar(Coleccion coleccion, Comparable dato)
    {
        return null;
    }
}

/**
 * Escriba el metodo buscar
 * @author
 * @version
 */
public class Binario extends Buscador
{
    public Comparable buscar(Coleccion coleccion, Comparable dato)
    {
        return null;
    }
}
```

7. *RTB* Los sistemas de recomendación de contenidos se han hecho muy populares en los últimos años. Entre los que se encuentran las recomendaciones de productos en tiendas online, películas, videos, música, libros, etc. Para hacer las recomendaciones, el sistema analiza y procesa información histórica de los usuarios.

El desafío de administrar las series televisivas (de elección libre similar a Netflix) y realizar recomendaciones sobre la base de las elecciones de los usuarios mediante una App. Cada vez que los usuarios ven una serie, el sistema registra la serie vista en un historial del usuario, para en un futuro puede hacerle recomendaciones la próxima vez que ingresa a la App.

Por cada serie se registra: título y categoría (acción, comedia, documental, etc).

En una primera etapa del proyecto las series recomendadas pertenecen a la misma categoría de la última serie vista.

Por ejemplo: Si se tiene un registro de series disponibles en la APP:

Big Ban	Comedia
Expedientes secretos	Suspense
Breaking Bad	Drama
Juego de Tronos	Drama
Sobrenatural	Terror
Friends	Comedia
Narcos	Drama

Si la última serie que vio el usuario fue Juego de Tronos, el sistema le recomendará, todas las series que tiene registradas en la categoría Drama por ejemplo: Breaking Bad, Narcos, etc.

- a) Diseña el diagrama de Clases que solucione al problema planteado
- b) Explica los relacionamientos de tu modelo
- c) El modelo debe tener atributos/métodos bien identificados
- d) Realiza un modelo que pueda en un futuro administrar los otros tipos de recursos como: música, libros, etc.

Capítulo 2

Recursividad

1. Encuentra un método que permita definir la potencia de un número. Los datos son **a** y **b** y lo que se desea es calcular **a** elevado a la **b**.
2. Escribe un proceso recursivo que permita decidir si un número natural **n** es primo o no.
3. Escribe un proceso recursivo que permita calcular la cantidad de dígitos que tiene un número positivo **n**.
4. Se desea encontrar los factores primos de un número **n**.
5. Diseña una clase **Secuencia** que represente una secuencia de números enteros positivos.
6. Añade a la clase **Secuencia**, un método que permita sumar los elementos de la secuencia.
7. Añade a la clase **Secuencia**, un método que permita encontrar el mínimo elemento de la secuencia.
8. Añade a la clase **Secuencia**, un método que permita ordenar los elementos de la secuencia.
9. Añade a la clase **Secuencia**, un método que permita buscar un elemento **x**.
10. Añade un método a la clase **Secuencia**, que permita aplicar un functor **f**. Un **Functor** es una clase de objetos que tiene la finalidad de especificar funciones que se realizarán sobre elementos enteros. Cuenta con las operaciones: **operar(int uno)** y **operar(int uno, int dos)**, ambas operaciones a su vez devuelven un **int**. Por ejemplo se podría tener el Functor **multi**, que básicamente, a través de sus métodos operar permiten multiplicar elementos del tipo **int**.
11. Añade un método a la clase **Secuencia**, que permita sumarlo con otra secuencia.
12. Bajo la suposición de que un objeto de la clase **Secuencia** se puede utilizar para representar un vector, añade un método a la clase **Secuencia**, que permita realizar producto escalar con otro vector.
13. Se desea sumar los números primos de una secuencia de números.
14. Escribe un método que permita saber si todos los dígitos de un número **n**, son pares.
15. Escribe un proceso recursivo que permita decidir si un número natural **n** es divisible entre 11. Dado que se sabe que un número es divisible entre 11, si y solo si la suma de los dígitos de posición par menos la suma de los dígitos de posición impar es un múltiplo de 11. Por ejemplo: sea **n** = 2341675, entonces $(5 + 6 + 4 + 2) - (7 + 1 + 3) = 6$, que no es múltiplo de 11, por lo tanto **n** no es divisible entre 11.

16. Se dice que **n** es un número que explota, cuando éste explota en varios fragmentos más chicos que él, dada una bomba. Si se tiene que **n** es el número y **b** la bomba, tales que **n** es mayor que **b**, se puede hacer que **n** explote en dos números $n1 = n / b$ y $n2 = n - (n/b)$. Pero **b** es una bomba que produce una reacción en cadena, si **n1** o **n2** son mayores que **b**, éstos también explotan con la regla anterior, hasta que se encuentre que el número no es mayor que **b**, entonces se dice que ya no se puede explotar el número. Por ejemplo, si **n = 10** y **b = 3**, entonces **n** explota en los pedazos: 3,2,1,1 y 3. Escribe un proceso que permita mostrar los pedazos del número **n**, dado que se tiene la bomba **b**.
17. Dado una cadena **c**, escribir procesos recursivos para resolver cada uno de los siguientes problemas:
- Contar la cantidad de veces que aparece un carácter **x** en **c**. Ej: para **c** = “elementos de programación” y **x** = ‘e’, el resultado es 4.
 - Generar una cadena **c1** equivalente a la cadena **c**, pero invertida. Ej: para **c** = “comer” el resultado es **c1** = “remoc”.
 - Contar la cantidad de vocales que tiene **c**. Ej: para **c** = “elementos de programacion”, el resultado es 10.
18. Definimos número promedio de un número entero positivo, al número que se obtiene de sumar sus dígitos de posiciones pares y restar sus dígitos de posiciones impares. Escribir un planteo recursivo para obtener el número promedio de un entero positivo dado. Ej.: el número promedio de 318547 es -2 pues $-3+1-8+5-4+7 = -2$.
19. Escribir una solución recursiva que calcule la función de Ackerman para valores de **m** y **n** dados. La función de Ackerman se describe del siguiente modo:
- $A(m, n) = n + 1$, si $m = 0$**
 $A(m, n) = A(m-1, 1)$, si $n = 0$
 $A(m, n) = A(m-1, A(m, n-1))$, si m es mayor que 0 y n es mayor que 0
20. Escribir una solución recursiva que retorne verdadero si una palabra es un palíndromo (una palabra es palíndromo cuando puede leerse igual de izquierda a derecha que de derecha a izquierda).
21. Implementar una solución recursiva que calcule la suma de los primeros **n** números pares que no son múltiplos de 4.
22. Escribir un proceso recursivo que muestre una media pirámide de dígitos como se muestra en la siguiente figura:
- ```
1
21
321
4321
54321
654321
7654321
87654321
987654321
```
23. Se define un **número de dígitos incrementales**, a todo número natural **n = dm ... d1 d0**, tal que **d<sub>i+1</sub> es menor igual que d<sub>i</sub>**. Por ejemplo: 1227, 359, 88, 139 son números de dígitos incrementales. Escribe un proceso recursivo para decidir si un número **n** es un número de dígitos incrementales.

24. Considera el anterior ejercicio, pero adáptalo para decidir si un número natural **n**, es un número de dígitos decrementales.
25. Definimos inverso simple de una secuencia de caracteres, al inverso de la secuencia sin sus repeticiones consecutivas. Por ejemplo, el inverso simple de **aaabccdddaaebb** es **beadcba**. Escribe un proceso recursivo para obtener el inverso simple de una secuencia de caracteres.
26. Escribir un proceso recursivo para reconocer si una cadena dada es de la forma **anbn**; es decir **n a's** consecutivas seguidas de **n b's** consecutivas.
27. Realizar un proceso recursivo que devuelva el **n-ésimo** elemento de menor valor en una secuencia de enteros positivos. Por ejemplo, si la secuencia contiene los valores 12, 7, 3, 9, 1, 5, 21 y **n = 1** devuelve 1, si **n = 2** devuelve 3, si **n = 5** devuelve 9 y si **n = 28** devuelve -1, que indica que ha habido un error.
28. Dadas dos cadenas **c1** y **c2**, diseñar procesos recursivos para:
- a) Decidir si **c1** es prefijo de **c2**. Ej.: prefijo("lo", "lombriz") = True; prefijo("árbol", "árboles") = False.
  - b) Decidir si **c1** es sufijo de **c2**. Ej.: sufijo("árbol", "teclado") = True; sufijo("soga", "mantel") = False
29. Escribir un proceso recursivo para determinar si un número positivo **n** es par, usando como pista la siguiente afirmación: Un número **n** es par si su predecesor es impar; un número **n** es impar si su predecesor es par.
30. Dado un número entero **n**, escribir un proceso recursivo para determinar cuántos dígitos pares ocupan posiciones impares en **n**. Ej.: para **n = 22005** el resultado es 2. Para **n = 201414**, el resultado es 1.
31. Escribir un proceso recursivo que dado un monto en pesos **m**, imprima el detalle de monedas y billetes para representar esa cantidad con la menor cantidad de billetes y monedas. Tener en cuenta que se tiene billetes de 100, 50, 20, 10, 5 y 2 pesos y monedas de 1 peso, 50, 25, 10, 5 y 1 centavo.
32. Dado un número entero positivo, se pide eliminar las segundas ocurrencias repetidas de dígitos.

Por ejemplo:

```
n = 2356342
el resultado sera
res = 56342
```

33. Dado un número entero positivo, se pide eliminar las primeras ocurrencias repetidas de dígitos.

Por ejemplo:

```
n = 2356342
el resultado sera
res = 23564
```

34. Encuentra la subsecuencia descendente más larga de una secuencia de elementos ordenables.

Por ejemplo:

```
sec = {2, 3, 5, 4, 3, 4, 2}
el resultado sera
res = {5, 4, 3}
```

35. Se pide encontrar la n-sima fila del triángulo de Pascal.

El triángulo de Pascal es el siguiente:

```

0 1
1 1 1
2 1 2 1
3 1 3 3 1
4 1 4 6 4 1

```

y asi sucesivamente.....

36. Se pide transponer un histograma ascendente armónico.

Por ejemplo:

```

*


```

el resultado sera

```

*
*
**


```

37. Rotar una onda en campana

Por ejemplo:

```



```

el resultado sera

```

*
**

**
*

```

38. Dada una secuencia de elementos se pide encontrar el conjunto clausura del mismo.

Por ejemplo:  
sec = {1,5,6}  
el resultado sera  
res = {{},{1},{5}, {6}, {1,5}, {1,6}, {5,6}, {1,5,6}}

39. Dada una cadena extendida se pide encontrar su formato reducido, el que indica el caracter de la cadena y la cantidad de veces que aparece.

Por ejemplo:  
cad = gggbdddeewwasss  
el resultado sera  
res = g3b2d2e3w2a1s3

40. Se pide eliminar las ocurrencias consecutivas impares de una cadena.

Por ejemplo:  
cad = aaabbabbaa  
el resultado sera  
res = aabbbbaa

41. Se pide eliminar las ocurrencias consecutivas pares de una cadena.

Por ejemplo:  
cad = aaabbabbaa  
el resultado sera  
res = ababa

42. Dada una cadena, se pide indicar cuántas veces existe un patron de forma exacta en la cadena original.

Por ejemplo:  
cad = aaabbabbaa  
patron = abba  
el resultado sera  
res = 2

43. Dada una cadena, se pide decidir si un patron se encuentra dentro de la misma por aproximación.

Por ejemplo:  
cad = aaabbabcaba  
patron = aba  
el resultado sera  
res = 2

Ya que existe por semejanza de caracteres en la subcadena abba y en aba, pero no asi en abca, ya que c genera la desigualdad

44. Se pide encontrar los  $n$  números primos de la secuencia de fibonacci.
45. Dado un texto se pide encontrar las palabras que ocurren en el mismo además del número de esas ocurrencias.

Por ejemplo:

```
cad = "lo que no me mata me fortalece"
el resultado sera
res = {("lo",1),("que",1),("no",1),
 ("me",2),("mata",1),("fortalece",1)}
```

46. Dado un tablero de ajedrez, ubique 8 reinas sin que ninguna de ellas se ataque, es decir corra peligro.
47. Se conoce que la mafia china es muy organizada y protege mucho a sus miembros, cuando deciden asistir a una reunión se dispone de una cantidad de chinos que asisten, y ellos se ubican de forma que al mirarlos frontalmente generan cierto respeto y temor. A continuación se tiene una serie de posibles reuniones y su nivel y la apariencia que se tiene del grupo que va a la reunión vistos frontalmente:

| Nivel reuni'on | Vista frontal de la delegaci'on |
|----------------|---------------------------------|
| 1              | (-.-)                           |
| 2              | (-.(.-.-).-)                    |
| 3              | (-.(.-.(.-.-).-).-)             |
| 4              | (-.(.-.(.-.(.-.-).-).-).-)      |

Considerando esta descripción, diseña los procesos recursivos que permitan mostrar la apariencia del grupo de chinos que asistena a una reunion de nivel  $n$ .

48. Se tiene una cadena *cad* y se quiere saber cual es la subcadena mas larga que existe en *cad* que respete un patron *pat*. Por ejemplo; si se tienen los siguientes casos de entrada:

```
Caso1 : cad es AAABBBBCCABCCBACBCBBAABBB
 pat es ABC
Caso2 : cad es AAABBBBABCBBB
 pat es ABC
Caso3 : cad es AAABBBB
 pat es ABC
Caso4 : cad es A
 pat es ABC
```

Las salidas serán:

```
Caso1 : 9
Caso2 : 3
Caso3 : 0
Caso4 : 0
```

49. Se tiene una cadena que representa una onda digital de se-  
nales L (Low) y H (High). Se pide encontrar la cantidad de ondas que respresenta.  
Por ejemplo si se tiene la cadena HHHLLLLHHHHHLLHLLHH, su onda digital se puede ver algo asi:

```

 --- | --- | --- | ---
 | | | | | | | | |

```

en esta linea se observan 5 ondas marcadas. Por lo que la respuesta es 5.

50. Se tiene una cadena que representa una onda digital de senales L (Low) y H (High). Se pide mostrar la onda que representa utilizando { \_ , | }, tal como se muestra en el siguiente ejemplo: si se tiene la cadena HHHLLLLHHHHLLHHLLHH, su onda digital se puede ver algo asi:

```

 --- | --- | --- | ---
 | | | | | | | | |

```

51. Se pide hacer el (los) método(s) recursivo(s) necesarios para rotar a derecha los elementos de una matriz en n posiciones a 0.

Si por ejemplo se tiene una matriz de 5x6

```

6 5 3 6 4 2
4 6 7 9 2 4
5 6 7 2 3 1
5 6 3 4 2 1
1 8 9 3 2 0

```

y un n de rotación 3 el resultado será:

```

6 4 2 6 5 3
2 4 4 6 7 9
1 5 6 7 2 3
5 6 3 4 2 1
1 8 9 3 2 0

```

Pero si se considera la matriz de 5x6

```

6 4 2 6 5 3
2 4 4 6 7 9
1 5 6 7 2 3
5 6 3 4 2 1
1 8 9 3 2 0

```

y el nivel de rotacion fuera 7, el resultado se obtiene en dos vueltas:

La primera:

```

3 6 4 2 6 5
2 4 4 6 7 9
5 6 7 2 3 1
3 4 2 1 5 6
3 2 0 1 8 9

```

La segunda:

6 5 3 6 4 2  
 9 2 4 4 6 7  
 5 6 7 2 3 1  
 3 4 2 1 5 6  
 3 2 0 1 8 9

52. Una red de mensajes en cadena ha encontrado una forma de repartir los mismos de manera que se pueda cubrir a la mayor gente posible.....Para ello han decidido usar una teoría, la misma que empieza enviando **X** mensajes, y a partir de este número se van enviando los mensajes de acuerdo a una teoría progresiva. Se termina de entregar los mensajes una vez que se encuentra 1. Sin embargo, no se sabe a ciencia cierta la teoría, lo único que se tiene son algunas fórmulas sueltas:

$$X/2 \quad y \quad X * 3 + 1.$$

y ejemplos de cómo se aplica:

CASO 1:

5, 16, 8, 4, 2, 1

CASO 2:

9, 28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

CASO 3:

10, 5, 16, 8, 4, 2, 1

CASO 4:

6, 3, 10, 5, 16, 8, 4, 2, 1

CASO 5:

13, 40, 20, 10, 5, 16, 8, 4, 2, 1

- a. En base a los casos deduce la teoría que se aplica para enviar los mensajes
- b. Escribe el (los) método(s) RECURSIVO(S) necesario(s) para saber cuántos mensajes se han enviado en total si se empieza con **X** mensajes. Este resultado se obtiene sumando la cantidad de mensajes sucesivos que se envían. Para dar un ejemplo: en el CASO 1: se envían 36 mensajes.
53. Se puede mostrar un número a través de una sucesión de funciones matemáticas de **sin** dado que se tiene las siguientes definiciones:

$$\text{Sea } A_n = \sin(1 - \sin(2 + \sin(3 - \sin(4 + \sin(n))))$$

$$\text{Sea } S_n = ((A_1 + n)A_2 + n - 1)A_3 + \dots + 2)A_n + 1$$

Dado un **n** imprimir  $S_n$

Se sabe que la entrada **n** es un número entero, tal que  $1 \leq n \leq 200$

Y la salida del proceso será una línea que contiene  $S_n$

Por ejemplo, si la entrada **n** es 3, entonces la salida es:  $((\sin(1) + 3)\sin(1 - \sin(2)) + 2)\sin(1 - \sin(2 + \sin(3))) + 1$

Escribe el (los) método(s) RECURSIVO(S) necesario(s) para dado un número entero **n**, permite mostrar  $S_n$

54. Se tiene un pequeño saltarin que camina sobre una cinta eléctrica hacia atrás o hacia adelante, esta cinta tiene etiquetada cada celda con una letra, el trabajo del saltarin es indicar los saltos y de que tamaño debería hacer para poder formar una palabra en base a sus saltos. Una condición es que debería saltar a



la posición más proxima válida para lograr su objetivo de formar la palabra. En caso de que se tengan dos opciones iguales se debe privilegiar el salto “adelante”.

Por ejemplo, si se tiene la cinta con las letras

$\{s, m, r, t, a, e, i, v, u, r, w, f, g, s, r, y, f, z, e, u, a, t, j\}$

Y el saltarin empieza en la posición 0 y se quiere formar la palabra *fuerza*, el saltarin debería dar los siguientes saltos: 12 adelante, 3 atrás, 3 atrás, 3 atrás, 15 adelante, 3 adelante.

Realiza el (los) método(s) RECURSIVOS necesarios para poder dar el reporte correcto del saltarin, en busca de su objetivo. Si no puede lograr el objetivo se debe emitir mensaje “IMPOSIBLE”.

55. Dada una secuencia de números enteros positivos, encuentre la secuencia de secuencias que permita tener tantos enteros como sean posibles de acuerdo al siguiente comportamiento:

CASO 1:

ENTRADA:

sec = {5, 16, 8, 4, 2, 1}

SALIDA:

secSec = {{1,2,3,4,5},  
          {1,2}}

CASO 2:

ENTRADA:

sec = {9, 28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10,  
      5, 16, 8, 4, 2, 1}

SALIDA:

secSec = {{1, 2, 3, 4, 5, 6, 7},  
          {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11},  
          {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17},  
          {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13},  
          {1, 2, 3, 4, 5},  
          {1, 2}}

CASO 3:

ENTRADA:

sec = {10, 7, 16, 8, 4, 2, 1}

SALIDA:

secSec = {{1, 2, 3, 4, 5, 6, 7},  
          {1, 2}}

CASO 4:

ENTRADA:

sec = {6, 10, 16, 8, 4, 1}

SALIDA:

secSec = {}

Escribe el (los) método(s) RECURSIVO(S) necesario(s) para obtener la secuencia de secuencias resultante, dado que se tiene una secuencia origen.

56. Un número es divisible por 7 cuando separando la primera cifra de la derecha, multiplicándola por 2, restando este producto de lo que queda a la izquierda y as sucesivamente, da cero o múltiplo de 7.

Por ejemplo, los siguientes números son divisibles entre 7, porque:

Ejemplo 1:

$$2'1 \quad 1*2 = 2$$

-2

===

0 es divisible entre 7

Ejemplo 2:

$$2633'4 \quad 4*2 = 8$$

-8

====

2625

$$262'5 \quad 5*2 = 10$$

-10

===

252

$$25'2 \quad 2*2 = 4$$

-4

===

21

$$2'1 \quad 1*2 = 2$$

-2

===

0 es divisible entre 7

Cuando la diferencia es mayor que 9 el proceso se repite.

Prueba este método con el ejemplo el 467, que no es divisible entre 7.

Define el (los) proceso(s) necesario(s) recursivo(s) para resolver si un numero n es divisible entre 7.

57. Se dice que un número es autocontenido básico de base n, cuando leído de izquierda a derecha este representa una secuencia ascendente de las potencias de n. Además se quiere saber la cantidad de autocontención que tiene el número, dado que la autocontención se define como la cantidad de potencias que existen en el número. En caso de que no sea un número autocontenido, la respuesta es -1.

Por ejemplo:

CASO 1:

numero = 1101001000100001000001000000

base = 10

La respuesta es 6 autocontenciones

CASO 2:

numero = 1392781

base = 3

La respuesta es 5 autocontenciones

CASO 3:

numero = 1248163254

base = 2

La respuesta es -1 autocontenciones

Realiza el(los) procesos RECURSIVOS necesarios para resolver el problema

58. Una entidad bancaria proporciona un número de identificación personal (un entero positivo) a cada uno de sus usuarios. El usuario de una tarjeta debe teclear su número de identificación para poder utilizarla.

Las terminales de la entidad traducen dicho número a una forma cifrada (también un entero positivo). Si ésta clave coincide con la forma cifrada grabada en la tarjeta, el usuario podrá utilizarla para llevar a cabo las operaciones deseadas.

Un número de identificación cuya representación decimal sea  $d_1 d_2 \dots d_n$  (donde  $n$  es el número de cifras y cada cifra  $d_i$  es un valor numérico comprendido entre 0 y 9; por ejemplo para 2145,  $n = 4$  y  $d_1 = 2, d_2 = 1, d_3 = 4, d_4 = 5$ ) se transforma en su correspondiente clave de la siguiente manera:

- Si  $n = 1$  la clave es  $d_1 * 7$ .
- Si  $n > 1$  la clave es  $(d_n \bmod 3) + d_n * c$  siendo  $c$  la clave de  $d_1 d_2 \dots d_{n-1}$ .

Se pide escribir el los métodos RECURSIVOS necesarios para que dado un número de identificación la transforme en su clave de identificación.

59. Debes escribir un programa que permita dada una cadena y un numero que indentifica el ancho de columna, partir esta cadena de tal manera que quepa en el ancho estipulado por la columna. Debes tratar de que las palabras no sean cortadas y debes justificar al ancho.

Como un procesador de textos, la ultima linea por supuesto no debe ser justificada y mas bien debe generar un retorno de linea.

**Ejemplo 1** Si se tiene como entrada:

Esta es una prueba de lo que se puede escribir en 100 espacios.

y el numero de columna es: 100

La salida es:

Esta es una prueba de lo que se puede escribir en 100 espacios.

**Ejemplo 2** Si se tiene como entrada:

Esta es una prueba de escribir un texto en columnas de 50 de ancho. Que es muy diferente a lo que se puede escribir en 100 espacios.

y el numero de columna es: 50

La salida es:

Esta es una prueba de escribir un texto en columnas de 50 de ancho. Que es muy diferente a lo que se puede escribir en 100 espacios.

Sobre la base de esta descripción se requiere que realices el (los) método(s) necesario(s) RECURSIVOS para resolver el problema.

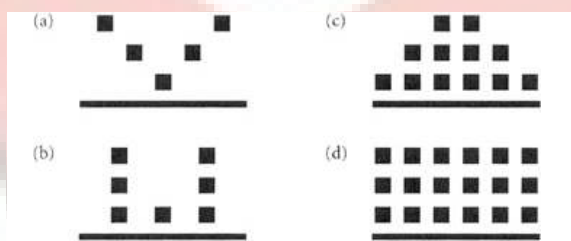
60. Se dice que un número es **siames**, si la mitad del número es igual a la otra mitad del número. Por ejemplo: 123123 es un número siames, pero 434 no lo es. La condición es que hagas el método **siames** recursivo, si tienes que usar otros métodos éstos también deben ser recursivos.
61. Escriba el(los) método(s) recursivo(s) necesario(s) para dado un numero de dos cifras  $n$ , se pueda dibujar ya sea: una V, un triángulo, una U o un rectángulo.

La cifra de las unidades de este número define el alto de la figura, y la cifra de las decenas define qué figura se tendrá. Si la cifra de las decenas es par y múltiplo de 4, entonces se dibuja una V, si es si es

impar y múltiplo de 3 se dibuja un triángulo, si es par, múltiplo de 2 y no de 4, entonces se dibuja una U, para el resto de los casos se dibuja un rectángulo.

Por ejemplo, considerando la figura 61 y si se tiene:

- el número 43, se dibujaría la figura de la opción a),
- en caso de que sea 93 se dibujaría lo que se muestra en el inciso c),
- si se tiene 63 se dibujaría lo que se tiene en b) y
- finalmente si se tiene 13 se dibuja lo que se muestra en inciso d).



Para realizar este dibujo puedes utilizar asteriscos. debes considerar los espacios que tienen estas figuras para verse bien.

62. Dado un número entero positivo, se pide conseguir un número que no contemple repetición de dígitos. Se debe eliminar las segundas ocurrencias de repetición de los dígitos.

Por ejemplo:

Si el número es:

$n = 345436456$

El número resultante será:

$res = 3456$

Realiza el (los) proceso(s) recursivo(s) necesario(s) para resolver el problema

63. Un número puede ser transformado para volver a ser el mismo. Para ello se toma el dígito de las unidades del número y se junta con el dígito de las unidades del número resultante de la suma de los dígitos del número original.

Por ejemplo, por ejemplo el 26 sufre de 4 transformaciones para volver a ser el mismo

| Inicio | Sume los dos dígitos | Combine los dos dígitos |
|--------|----------------------|-------------------------|
| 26:    | 2+6=08:              | "6"y"8"=68              |
| 68:    | 6+8=14:              | "8"y"4"=84              |
| 84:    | 8+4=12:              | "4"y"2"=42              |
| 42:    | 4+2=06:              | "2"y"6"=26              |

Realiza el (los) proceso recursivos necesarios para dado un número entero positivo entre 0 y 99 inclusive, indique la cantidad de transformaciones que se requieren para poder volver a generar el número original.

64. En un país de los acertijos, se tiene definida una fórmula secreta de construcción de paredes que permite administrar de mejor forma los ladrillos que se tienen para construir las paredes. Antes de construir se requiere saber la cantidad de paredes que se pueden hacer en base a los ladrillos que se tienen y todas sus posibilidades. La condición es tratar de hacer paredes cuadradas. Por ejemplo si se tienen 1234 ladrillos de  $1 \times 1$ , se pueden hacer las siguientes opciones:

(1 de  $35 \times 35$ ), (1 de  $3 \times 3$ );

(1 de  $34 \times 34$ ), (1 de  $8 \times 8$ ), (1 de  $3 \times 3$ ), (1 de  $2 \times 2$ ), (1 de  $1 \times 1$ );

(1 de  $33 \times 33$ ), (1 de  $12 \times 12$ ), (1 de  $1 \times 1$ );

y así sucesivamente

No se quiere tener dos paredes de la misma dimensión.

Entonces se tiene como entrada el número de ladrillos y las dimensiones de los ladrillos  $l$  y  $a$  y en base a las mismas, se quiere obtener las secuencias posibles de paredes que se pueden construir. Realiza el (los) método(s) recursivo(s) necesario(s) para obtener las distintas opciones de construcción.

65. Hacer el(los) método(s) recursivo(s) necesario(s), para calcular la raíz cúbica de un número de la siguiente manera:

Si  $N=1$  el cubo se obtiene de  $1 = 1$

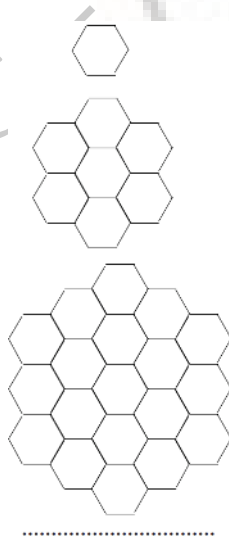
Si  $N=2$  el cubo se obtiene de  $3 + 5 = 8$

Si  $N=3$  el cubo se obtiene de  $7 + 9 + 11 = 27$

Si  $N=4$  el cubo se obtiene de  $13 + 15 + 17 + 19 = 64$

y así sucesivamente.

66. Se pide diseñar proceso(s) recursivo(s) necesario(s) para encontrar la cantidad de líneas que se requieren para construir hexágonos de crecimiento de nivel  $n$ , como se muestra en la figura 66.



Para nivel 1 se requieren 6 líneas

Para nivel 2 se requieren 30 líneas

Para nivel 3 se requieren 72 líneas

67. Se dice que una frase es un **Tautograma** si las palabras que están en la frase comienzan con la misma letra.

Caso 1

Si la frase es:

Flowers Flourish from France

La respuesta es verdad

Caso 2

Si la frase es:

Sam Simmonds speaks softly

La respuesta es verdad

Caso 3

Si la frase es:

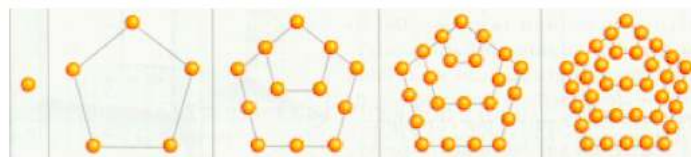
Peter EpIckEd PePPers

La respuesta es falso

Escribe el (los) método(s) recursivo(s) necesario(s) para dada una frase, decidir si es o no **tautograma**.

68. Dentro el mundo matemático hay muchas relaciones numerales interesantes una de ellas son los numeros poligonales.

Se pueden formar pentágonos de distintos niveles de crecimiento, juntando círculos como muestra en la Figura 2.1. Dada esta relación se pide escribir el (los) proceso(s) necesarios RECURSIVOS para encontrar la cantidad de círculos necesarios para tener un pentágono de nivel n.



|                 |          |          |          |          |
|-----------------|----------|----------|----------|----------|
| <b>Nivel:</b> 1 | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> |
| 1               | 5        | 12       | 22       | 35       |

Figura 2.1: Número pentagonal

69. En el país de los acertijos, nuestro matemático reconocido **Ibn Albania**, ha encontrado una relación interesante entre números y operaciones aritméticas. Entre ellas una de las más curiosas es la siguiente

$$9 \times 9 + 7 = 88$$

$$98 \times 9 + 6 = 888$$

$$987 \times 9 + 5 = 8888$$

$$9876 \times 9 + 4 = 88888$$

$$98765 \times 9 + 3 = 888888$$

.....

Y así sucesivamente

En base a esta relación, se desea dado un número entero de entrada encontrar su equivalente en terminos de la operación de la multiplicación y suma si es posible.

Por ejemplo,

ENTRADA: 8

SALIDA: "No es posible"

ENTRADA: 8888

SALIDA: "987 x 9 + 5"

ENTRADA: 675

SALIDA: "No es posible"

Realiza el(los) método(s) RECURSIVO(S) necesario(s) para resolver el problema.

70. En la química orgánica se tienen cadenas de Carbono e Hidrógenos juntos, que forman distintas clases de alcanos; que se diferencian unos de otros por la cantidad de carbonos que existen. La forma de encontrar además la nominación correcta es considerando la cadena lineal más larga que se pueda tener, la misma que da la nominación básica y encima de esa se obtienen otras más, que se basan en las ramificaciones existentes sobre la cadena lineal principal. Lo importante por el momento es identificar la cadena más larga lineal de carbono que se pueda hallar en una composición. Por ejemplo si se tiene, la composición:

```
. C
C C . C
. C . C C .
. C . C C .
. C C C C C C C C .
. C
. C
```

donde el C, significa la existencia de Carbono y . la no existencia de Carbono, se tiene que la cadena lineal más larga es de 15 Carbonos.

Se pide, dada una composición - que es una matriz de símbolos {C,.}, encontrar la cadena lineal más larga de carbonos; para ello debes considerar tu solución de forma RECURSIVA.

71. Se desea hacer proceso(s) recursivo(s) necesario(s), para poder eliminar las ocurrencias consecutivas de un mismo carácter siempre y cuando sea ocurrencias impares, por ejemplo: si se tiene **aaa** se reduce a **aa**, en cambio las **a** se reduce a la cadena vacía. Por supuesto si se tiene la cadena **aabab** esta se reduce a **aa**. Por lo tanto, la cadena resultante puede tener más de una ocurrencia consecutiva de un carácter siempre y cuando sea en una cantidad par. A continuación, se muestran dos ejemplos más complejos con sus soluciones:

Ejemplo 1 Entrada:

aabbbaaaaccaabbccacda

Salida:

aabbbaaaaccaabbcc

Ejemplo 2 Entrada:

bbbdhshsgggyrrmmddee

Salida:

bbhhggyrrmmddee

72. Dado un conjunto de enteros positivos desordenados, se pide encontrar la suma de ascendientes más grande que existe en la secuencia. Por ejemplo si se tiene la siguiente secuencia: 1, 4, 3, 5, 36, 2, 6, 8, 9, 3, 2, 4, 27. Se tienen las secuencias ascendientes siguientes: (1, 4), (3, 5, 36), (2, 6, 8, 9), (3), (2, 4, 27), la respuesta debiera ser 44. Realiza el (los) método(s) RECURSIVOS necesario(s) que permitan resolver el problema.



73. Realizar el (los) proceso(s) recursivos(s) que devuelva(n) el primer término de la serie de números primos que cumpla que tiene “n” dígitos. El parámetro del proceso es “n”.
74. Dado un conjunto de números, se desea encontrar el conjunto clausura = n. A continuación se tienen tres casos como ejemplo:

**Ej1.** Dado el conjunto  $A = \{3, 5, 6, 2, 8, 10, 23, 9\}$ , se quiere encontrar el conjunto clausura = 2, el resultado seria:

$$A^2 = \{\{3, 5\}, \{3, 6\}, \{3, 2\}, \{3, 8\}, \{3, 10\}, \{3, 23\}, \{3, 9\}, \{5, 6\}, \{5, 2\}, \{5, 8\}, \{5, 10\}, \{5, 23\}, \{5, 9\}, \{6, 2\}, \{6, 8\}, \{6, 10\}, \{6, 23\}, \{6, 9\}, \{2, 8\}, \{2, 10\}, \{2, 23\}, \{2, 9\}, \{8, 10\}, \{8, 23\}, \{8, 9\}, \{10, 23\}, \{10, 9\}, \{23, 9\}\}$$

**Ej2** Dado un conjunto  $A = \{4, 3, 8, 10, 34\}$ , se quiere encontrar el conjunto clausura = 3, el resultado seria:

$$A^3 = \{\{4, 3, 8\}, \{4, 3, 10\}, \{4, 3, 34\}, \{3, 8, 10\}, \{3, 8, 34\}, \{8, 10, 34\}\}$$

**Ej3** Dado un conjunto  $A = \{5, 6, 2, 3\}$ , se quiere encontrar el conjunto clausura = 6, el resultado seria:  
 $A^6 = \{\}$

Escribe el (los) método(s) **RECURSIVOS** necesario(s) para encontrar el conjunto clausura = n, dado un conjunto de enteros.

75. Se dice que un número es autocontenido básico de base n, cuando leído de izquierda a derecha este representa una secuencia ascendente de las potencias de n. Además se quiere saber la cantidad de autocontención que tiene el número, dado que la autocontención se define como la cantidad de potencias que existen en el número. En caso de que no sea un número autocontenido, la respuesta es -1.

Por ejemplo:

CASO 1:

numero = 1101001000100001000001000000

base = 10

La respuesta es 7 autocontenciones

CASO 2:

numero = 1392781

base = 3

La respuesta es 5 autocontenciones

CASO 3:

numero = 1248163254

base = 2

La respuesta es -1 autocontenciones

Realiza el(los) procesos **RECURSIVOS** necesarios para resolver el problema

76. Un mafioso muy conocido ha decidido irse a vivir a una gran ciudad, en la que tiene una gran cantidad de familia. Cada familiar vive sobre una misma avenida en distintas calles las mismas que están numeradas. Don Vito, quiere ubicarse en la ciudad y quiere minimizar la distancia de su casa a las casas de sus familiares. Escribe un programa que ayude a Don Vito a encontrar la mejor ubicación dado que tiene las direcciones de sus parientes y además la menor distancia que recorrerá si visita de casa en casa de acuerdo a una secuencia dada.

Por ejemplo si tiene 6 familiares y sus casas están en las calles: 2, 5, 3, 7, 4, 2. La mejor ubicación para comprarse una casa es: la calle 3, a que tiene la suma de la distancia mas óptima de llegar a sus familiares.

Realiza el (los) método(s) necesario(s) **RECURSIVO(S)** para resolver el dilema de Don Vito.



77. Realiza el (los) método(s) **RECURSIVO(S)** necesarios para poder calcular los números primos menores que  $N$  con el algoritmo de la **Criba de Eratóstenes**. La Criba de Eratostenes es un procedimiento que permite hallar todos los números primos menores que un número natural dado  $N$ . Se forma una tabla con todos los números naturales comprendidos entre 2 y  $N$  y se van tachando los números que no son primos de la siguiente manera: cuando se encuentra un entero que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos. El proceso termina cuando el cuadrado del mayor número confirmado como primo es mayor que  $N$ .

Ejemplo:

Determinemos, mediante este procedimiento, la lista de los números primos menores de 20.

Para ello definiremos el \* como "marcado" y el - como "tachado".

Sea  $N = 20$

Primer paso:

Escribamos la lista de los números naturales comprendidos entre 2 y 20.

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,  
16, 17, 18, 19, 20

Segundo paso:

Marcamos el primer número no tachado ni marcado como número primo.

2\*, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,  
16, 17, 18, 19, 20

Tercer paso:

Tachamos todos los múltiplos del número que acabamos de marcar como primo.

2\*, 3, 4-, 5, 6-, 7, 8-, 9, 10-, 11, 12-, 13, 14-, 15,  
16-, 17, 18-, 19, 20-

Cuarto paso:

Si el cuadrado del número recién marcado es inferior a 20, entonces repetimos el segundo paso. Si no, el proceso termina, y todos los enteros no tachados son declarados primos.

Como  $3^2 = 9 < 20$ , volvemos al segundo paso:

2\*, 3\*, 4-, 5, 6-, 7, 8-, 9-, 10-, 11, 12-, 13, 14-, 15-,  
16-, 17, 18-, 19, 20-

En el cuarto paso, el primer número que no ha sido tachado ni marcado es 5. Como su cuadrado es mayor que 20, el proceso termina y consideraremos primos todos los números que no han sido tachados.

RESULTADO: Los números primos comprendidos entre 2 y 20 son:

2, 3, 5, 7, 11, 13, 17, 19.

Se pide diseñar proceso(s) recursivo(s) necesario(s) para decidir si un número es un super número. Se dice que un número es un super número, si y solo si la suma de sus dígitos al cuadrado, es menor que la suma de los dígitos mas significativos (excluyendo el dígito de la unidad) al cubo y esta suma a su vez es menor que la suma de los dígitos mas significativos (excluyendo las decenas) elevadas a la cuarta, y así sucesivamente.

Caso 1:

$n = 111111$

no es super número ya que la suma de sus dígitos al cuadrado 6 es mayor a la suma de los cubos del número sin unidades que es 5.

Caso 2:

$n = 55432$

si es super número, ya que la suma de los dígitos al cuadrado es 79 y menor que la suma de los dígitos al cubo que es 341, y 341 es menor que la suma de los dígitos mas significativos excluyendo decenas elevado a la cuarta que es 1506.

1506 a su vez es menor que 6250 que vendría a ser la suma de los dígitos a la 5 excluyendo las centenas.

Y por ultimo 6250 es menor a 15625.

78. En la selva se han mimetizado varios leones, grrrrr....., debemos pasar la selva.... es necesario.... pero no queremos morir..... :S, - piensa Hugo; pero como debe guardar compostura y mantener la calma con sus hermanos Paco y Luis; no muestra su temor. Mas al contrario ha decidido buscar cuántos leones hay mimetizados para saber el número de balas que debe cargar si es necesario tomar alguna medida de emergencia.

Pero hay un problema y es que la selva de letras que deben atravesar puede ser muy tupida y complicada; por lo que ha decidido recordar sus dotes de programador y no ha tenido éxito. Por lo que ha solicitado que escribas un programa RECURSIVO para encontrar el número de leones, dado que se tiene la selva de letras.

Por ejemplo, si se tiene la selva:

SELVADELEJEMPLODONDESEENCUNETRANDOSLEONES

La respuesta será 2.

Sin embargo si se tiene la siguiente selva:

LEONIDASUNGRANGUERRERODELAANTIGUAGRECIA

La respuesta será 1.

79. Buscando el nido especial están dos pajaritos. Pues dirás cuál es el problema?, y la respuesta es que estos pajaritos tienen requerimientos muy exquisitos del lugar donde quieren vivir. Ellos son dos pájaritos de cuello azul y les gustaría vivir en la zona que haya más pajaritos azules y por supuesto siempre y cuando exista un nidito libre.

Para ello la pajarita ha conseguido un plano de la ubicación de los nidos libres y los vecinos que circundan esos nidos. El plano es rectangular en el que los nidos libres se especifican con un \* y la cualidad de

los pajaritos con un letra, que es la inicial del color del pajarito vecino, es así que: **A** especifica al color “AZUL”, **V** al “VERDE” y así..... debes considerar que no es posible que dos colores distintos sean explicados con la misma letra.

Debes ayudar a estos pajaritos a hallar su nido perfecto o casi perfecto!!!

Por ejemplo, si se tiene el siguiente plano de posibilidades:

```

A A A * V V V V N N N N * N N
A A V V V * V V V V V V A A A
A A A A A A A A A A A A A A
A A A A A A A A A A A A A A
V V V V V V V V V V V V V V
V V V V V V V * V V V V A A A
V V V * V V V V V V V V A A A
V V V V V V V V V V V V A A A
A A A A A A A V A A A V A A A
A A A A A * A A V A A V A A A
A A A V A A A V A A A V A A A
A A A A A A A V A A V V * V
V V V A A A A V A V V V A A A

```

Los pajaritos con esta información han decidido irse a vivir al nido ubicado en (1, 4) ya que la comunidad de sus congéneres a los que puede llegar es la mayor, que con los otros nidos. En caso de que haya más de un nido que cumpla la condición, debes responder la ubicación del primer nido que hayas encontrado.

Como habrás podido notar, a los pajaritos les parece mejor vivir en la “orilla” de una zona de sus congéneres, si eso les da mayor posibilidad de relación con los de su clase, que vivir al medio de los de su clase con un círculo social menor. Es que.... en resumidas cuentas la pajarita quiere hacer mucha vida social.

80. Considera el problema de los pajaritos, pero ahora se ha añadido un dato más y es la información del número de integrantes de cada familia, entonces, **A,3**, significa que en esa posición vive una familia de pájaros azules con tres integrantes.

Se te pide encontrar cual es el nido correcto para esta pareja de pajaritos, si ahora se tiene la siguiente información de la región en la cual piensan comprar su nidito.

```

A,2 A,1 A,1 *,0 V,1 V,1 V,1 V,1 V,1 N,2 N,3 N,2 N,2 *,0 N,1 N,1
A,1 A,1 V,1 V,1 V,1 *,0 V,1 V,2 V,3 V,1 V,1 V,1 V,1 A,5 A,7 A,6
A,2 A,2 A,2 A,2 A,3 A,7 A,3 A,3 A,8 A,3 A,3 A,3 A,2 A,2 A,2 A,2
A,1 A,1 A,2 A,3 A,4 A,5 A,1 A,4 A,5 A,9 A,2 A,3 A,2 A,2 A,4 A,5
V,2 V,1 V,1 V,1 V,1 V,2 V,2 V,2 V,6 V,7 V,8 V,5 V,6 V,4 V,5 V,4
V,1 V,2 V,4 V,3 V,2 V,3 V,2 V,6 *,0 V,3 V,3 V,2 V,2 A,1 A,1 A,1
V,1 V,1 V,2 *,0 V,1 V,5 V,3 V,3 V,2 V,1 V,1 V,1 V,4 A,1 A,1 A,1
V,2 V,3 V,4 V,5 V,5 V,2 V,2 V,1 V,4 V,5 V,7 V,8 V,3 A,1 A,1 A,1
A,1 A,1 A,1 A,1 A,1 A,1 A,1 V,3 A,1 A,1 A,4 V,1 A,2 A,1 A,1
A,2 A,2 A,2 A,2 A,2 *,0 A,2 A,1 V,2 A,1 A,2 A,3 V,1 A,1 A,2 A,1
A,2 A,3 A,1 V,1 A,2 A,1 A,2 A,1 V,5 A,2 A,2 A,1 V,5 A,1 A,1 A,1
A,2 A,1 A,1 A,1 A,1 A,1 A,1 V,1 A,1 A,1 V,1 V,1 V,1 *,0 V,1
V,1 V,1 V,1 A,1 A,1 A,1 A,1 V,1 A,1 V,1 V,1 V,1 A,1 A,1 A,1

```



Donde O y G significan que son elementos “impasables” y el resto de las letras son elementos “pasables”.

Y la posición del PACMAN es (3,9)....

El PACMAN, podría comer frutas y objetos por arriba con “bonos”  $5000 + 1000$  haciendo un total de 6000, por la derecha  $1000 + 700 + 5000 + 5000$  con un total de 11700, por abajo come 0 y por la izquierda  $100 + 100 + 500 + 300$  con un total de 1000. Como tiene solo dos posibilidades se elige arriba y la derecha haciendo un total de 17700 “bonos”.

Puedes considerar: que jamás el PACMAN puede andar en círculos, que se tiene el plano del juego y la posición inicial del PACMAN.

Ya se ha adelantado algo de la solución y se tiene el modelo de la Figura 3.1:

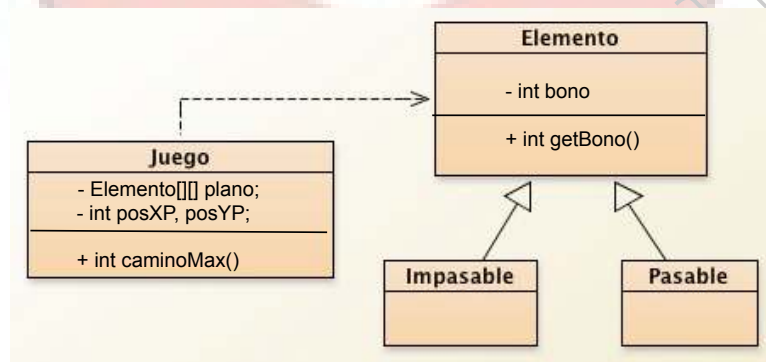


Figura 2.2: Modelo

Se te pide sobre la base de este modelo estructurar tu solución, utilizando recursividad. Estas en libertad de escribir los métodos que consideres necesarios adicionales a lo que se tiene en el modelo.

84. En el país de las letras existen unas muy juguetonas que se encuentran escondidas en su calificativo. Por ejemplo: la palabra “lirica” no es juguetona ya que no contiene todas las letras de su calificativo “jugueton”, es más la pobre palabra es muy seria :(. Sin embargo, la palabra “jeringuito” es una juguetona palabra, otra palabra juguetona es: “esterojiligrantapus” o “supermegamentojaso”

Tu tarea es decidir dada una palabra si es o no una palabra “juguetona”

85. Un virus muy feroz ha atacado la computadora de Lulú, y ha dañado algunos documentos que tenía, como es previsora ella tiene un backup de su información, y es posible que recupere sus documentos. Sin embargo, inquieta como es se ha dado a la tarea de averiguar qué cambios ha hecho el virus en los documentos, y recopilar en orden las letras que ha eliminado el virus comparando con su documento del backup. Al poco tiempo se ha dado cuenta que es tedioso y sería interesante construir un programa que le permita hacer este trabajo.

A continuación te brindamos un ejemplo reducido de la situación que Lulú tiene:

st s una soria de mieo, las ltas dsaparcn in abero y son asenadas r

esta es una historia de misterio, las letras desaparecen sin saberlo y son asesinadas por

donde en primer texto es el dañado por el virus, el segundo es el que tiene en su backup

Entonces la cadena de las letras que el virus ha eliminado es:

eaehitteriereeeesslesipo

Realiza el (los) procesos recursivos necesarios, para que dado un texto dañado, un texto recuperado del backup; indique las letras que han sido eliminadas por el virus.

86. En la guerra de los “**Kbots**” se han diseñado robots que permiten explorar caminos seguros en campos minados. Los humanos deben caminar por este campo minado buscando documentos secretos que están ubicados en algún lugar del campo. La misión de estos robots es indicar cuál es el camino más seguro, además de informar en qué posiciones están las minas para que los exploradores humanos no las pisen y puedan cumplir con su objetivo.

En este campo minado existen alambres de púas que no pueden ser sobrepasados, espacios seguros de camino, minas, el lugar donde están los exploradores humanos y el lugar donde se encuentran los documentos secretos.

Por ejemplo:

PPPPMPMM  
PESSSMSSP  
SSSPPPSSS  
SMSSPPMSM  
SSSSDSSMS  
PPSSSSMSS  
SSMSMSMSM

Donde E representa la posición en la que los exploradores están, P identifica la existencia de púas, M indica la existencia de una mina y D indica el lugar donde están los documentos secretos.

Los robots sólo se mueven en línea recta, es decir jamás diagonal, considerando estas condiciones encuentra el camino más seguro de exploración.

Tomando en cuenta el ejemplo anterior, el camino más seguro será:

(R,S), (D,S), (D,S), (D,S), (R,S), (R,S), (R,D)

En este caso se ha encontrado un camino sin minas, pero en caso de que no exista tal camino, se debe devolver el camino que tiene menos minas e informar dónde están para que los exploradores tengan cuidado.

Para aclarar, en el camino se tiene una lista de pasos que deben dar, cada uno de estos pasos es de la forma (DIRECCION, SITUACION), por ejemplo (R,S) dice que el movimiento es a la derecha=RIGHT y la situación es segura=S. (Para completar la dirección puede ser: R=derecha, L=izquierda, D=abajo, U=arriba; la situación puede ser: M=mina, D=documento, S=segura)

Realiza el (los) procesos recursivos necesarios, para que dado un campo minado, la ubicación de los exploradores, la ubicación de los documentos; se encuentre el camino “más” seguro de exploración.

87. En **NumeroLandia**, existen diferentes ciudadanos y entre ellos existen los curiosos, éstos cumplen una condición interesante y es que la suma del factorial de sus dígitos da el mismo número, por ejemplo 145 es un número curioso, pero el 77 no es número curioso.

Realiza el (los) procesos recursivos necesarios, para que dado un número, indique si es o no curioso.

88. En la toma de exámenes se ha detectado que hay dos personas que siempre andan juntas e intentan sentarse juntas, esto es por demás sospechoso para los profesores. Por lo que se te ha dado la tarea de evitar que estas dos personas se sienten juntas, para lo cual te dan como datos una lista de estudiantes y los nombres de las dos personas en cuestion.

Por ejemplo, si se tiene:

```

luis
orian
darla
tom
orian darla

```

Los primeros 4 nombres corresponden a la lista de los estudiantes, y la ultima fila tiene los nombres de las personas sospechosas. Con esta información se debe obtener todas las posibles conformaciones de ubicacion de los estudiantes de tal manera que los sospechosos jamas queden juntos.

La respuesta se veria asi:

```

luis orian tom darla
luis darla tom orian
orian luis darla tom
orian luis tom darla
orian tom luis darla
orian tom darla luis
darla luis orian tom
darla luis tom orian
darla tom luis orian
darla tom orian luis
tom orian luis darla
tom darla luis orian

```

Tu tarea es encontrar una solución RECURSIVA que permita resolver este problema, dado que se da una lista de nombres y un par de nombres que no se pueden sentar juntos.

89. Se tiene un plano de las zonas rurales de nuestro departamento a los cuales llegará la señal del satélite, considerando el espectro de alcance del satelite denotado por A. Se requiere desarrollar un programa RECURSIVO que permita decidir a cuanta población beneficiara el satélite si esta ubicado en la coordenadas X e Y y tiene un alcance A.

La figura 4.12 el inciso a). muestra un ejemplo del plano y de la densidad poblacional de las areas, y el inciso b). muestra cuales serian las zonas a las que el satélite alcanzaría si estuviera ubicado en la posición (4,5) y su alcance fuera de 3; el resultado que es la población beneficiada será 8126

90. Se desea saber si un numero es monticulo, se dice que los es si imaginariamente tu puedes formar una onda o un pico creciente-decreciente, con los valores del numero. Considerando cada digito como valor. Por ejemplo el numero 12346321 es un monticulo, ya que se puede formar:

```

*
*
**


```



|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 45  | 0   | 100 | 78  | 0   | 0   | 0   | 45  |
| 0   | 200 | 900 | 0   | 32  | 48  | 86  | 0   |
| 230 | 0   | 700 | 0   | 0   | 0   | 0   | 0   |
| 0   | 0   | 0   | 768 | 943 | 0   | 577 | 0   |
| 234 | 56  | 0   | 98  | 0   | 564 | 234 | 564 |
| 23  | 634 | 76  | 82  | 34  | 12  | 0   | 0   |
| 0   | 200 | 0   | 0   | 0   | 457 | 0   | 892 |
| 0   | 200 | 893 | 0   | 32  | 48  | 86  | 0   |
| 23  | 634 | 76  | 82  | 34  | 12  | 0   | 0   |

a). Plano de areas pobladas y su población

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 45  | 0   | 100 | 78  | 0   | 0   | 0   | 45  |
| 0   | 200 | 900 | 0   | 32  | 48  | 86  | 0   |
| 230 | 0   | 700 | 0   | 0   | 0   | 0   | 0   |
| 0   | 0   | 0   | 768 | 943 | 0   | 577 | 0   |
| 234 | 56  | 0   | 98  | 0   | 564 | 234 | 564 |
| 23  | 634 | 76  | 82  | 34  | 12  | 0   | 0   |
| 0   | 200 | 0   | 0   | 0   | 457 | 0   | 892 |
| 0   | 200 | 893 | 0   | 32  | 48  | 86  | 0   |
| 23  | 634 | 76  | 82  | 34  | 12  | 0   | 0   |

b). Alcance del satellite con A = 3 y  
ubicación X = 4, Y = 5. Población  
beneficiada es 8126

Figura 2.3: Ejemplo del problema

Si embargo el numero 12342345, no lo es porque se puede apreciar mas de un pico:

```

 *
 * **
 ** ***


```

Realiza un programa recursivo que dado un numero me permita decidir si es monticulo o no.

91. Caperucita roja esta en el bosque, camino a casa de su abuelita; ella en los últimos años se hizo muy amiga del lobo, pero ahora tiene miedo de otros animales que suelen vivir en cabañas. Horror!!! cuánto ha cambiado el mundo, lo que ella quiere es llegar a la casa de su abuelita con la mayor cantidad de flores que pueda recoger. Ayudale a decidir cuántas flores puede recoger.

Ella actualmente debe recorrer el camino a la casa de su abuelita y recoger la mayor cantidad de flores, cada que ve una cabaña ella jamas se acerca a ella aunque allí hayan flores, si se topa con el lobo, lo saluda cortezmente y continua recogiendo flores.

El terreno que debe recorrer la caperucita se puede planear como un plano rectangular en el que se encuentran los distintos elementos antes descritos: Flores(F), Via libre (V), Lobo(L), Cabaña( C ) y Casa de abuelita(A). Caperucita siempre empieza su recorrido en la posición 0,0.

Un ejemplo de recogida de flores seria:

```

VVVVVFFFCCCCC
VVVVVCCCCCCCCC
CCVVVVVVVVLFFF
CCVVVVVVVVVFFF
CCCCCVCVVVFFF
CCFFFFCCCVVFCC
CCFFCCCCVVVACC
CCCCCCCCFFFFF

```

La respuesta es 22 flores.

Escribe un programa recursivo que ayude a caperucita roja llenar su cesta con la mayor cantidad de flores, debes devolver el numero de flores recolectada.



92. En **NumeroLandia**, hay variedad de numeros que cumplen ciertas cualidades, no podian faltar los números sociables.

Se dice que varios numeros son sociables si: La suma de los divisores del primer número da el segundo, la suma de los del segundo da el tercero, y as sucesivamente. La suma de los divisores del último da el primer número de la lista.

Por ejemplo los números 12496, 14288, 15472, 14536 y 14264 son números sociables. Ya que: los divisores de: 12496; son: 1, 2, 4, 8, 11, 16, 22, 44, 71, 88, 142, 176, 284, 568, 781, 1136, 1562, 3124, 6248 y su suma es 14288. Del mismo modo los divisores de 14288 son: 1, 2, 4, 8, 16, 19, 38, 47, 76, 94, 152, 188, 304, 376, 752, 893, 1786, 3572, 7144 y su suma es 15472. A su vez los divisores de 15472 son: 1, 2, 4, 8, 16, 967, 1934, 3868, 7736 que sumados dan 14536. Los divisores de 14536 son: 1, 2, 4, 8, 23, 46, 79, 92, 158, 184, 316, 632, 1817, 3634, 7268 y su suma es 14264. Finalmente los divisores de 14264 son: 1, 2, 4, 8, 1783, 3566, 7132 que suman 12496, que vendría a cerrar el círculo de amistad.

Dado un conjunto de números determina si son sociables o no, considerando la descripción del problema. Para ello debes escribir un programa recursivo

93. <sup>1</sup> La vida no es ningún pasillo recto y fácil que recorremos libres y sin obstáculos, sino un laberinto de pasadizos, en el que tenemos que buscar nuestro camino, perdidos y confusos, detenidos de vez en cuando, por un callejón sin salida. Esto lo entienden Hem y Haw dos homrecitos atrapados en un laberinto, donde existen minas de queso, algunas de ellas contienen quesos frescos, deliciosos y nutritivos (F), otros quesos podridos y a veces mortales (M), también existen pasadillos libres (L) que no contienen nada pero es posible transitar por ellos, las paredes (P) que no se pueden atravesar y por supuesto a salida secreta (S).

Ayudemos a encontrar la salida a Hem y Haw, diseñando una solución recursiva, el mismo que debe además indicar cuantas minas de queso fresco encontraron en el camino mas corto, como comprenderas ellos no estan interesados en encontrar minas de queso sino más bien salir del laberinto.

Un ejemplo de laberinto es:

```
L P L M P P P L
L F F P L P F L
L P P M L P P L
L F F F F F F L
P P P L L P P L
P L P P L P P P
M L F F L F F F
P F P P L P P P
F F P F F F F F
P P P F L M L P
P L L F L P P P
F F L F F F F F
P L L L F P S P
P L P P F M P P
P L M P F L L L
```

Además se tiene la posición inicial de Hem y Haw que denotaremos posIX y posIY. Por otro lado se tiene la posicion de la salida secreta salX y salY. Hem y Haw no pueden saltar las paredes y tampoco caminar en diagonal. Nunca Hem y Haw inician en una posición que no sea una L.

<sup>1</sup>Propuesto por Lic. Rossmery Torrico

94. Se tiene un poema escrito con mucha matemática, que es eso? te preguntará y la respuesta es que dado un número (usualmente de trascendencia matemática como  $\pi$  o  $e$  o algún otro)  $n$ , y un poema debes decidir si es un poema matemático de  $n$

Por ejemplo: si  $n$  es 31415926535897932384 y el poema es:

Soy y sere a todos definible.  
Mi nombre tengo que daros;  
cociente diametral siempre inmedible  
soy de los redondos aros

El poema cumple con ser de naturaleza matemática 31415926535897932384

Pero si  $n$  es 31415926535897932 y el poema es:

Soy P ella I somos camaradas.  
La quiero mucho, que si descuido  
cerebrito nuestro pequenito rey  $\pi$ .

El poema cumple con ser de naturaleza matemática 31415926535897932.

Pero si  $n$  es 2718281828459 (que es el valor de  $e$ ) y el poema es:

Soy  $e$  valor aureo,  
como el sol y las estrellas.  
Trascendente en varias  
formulas de conteo.

El poema NO cumple con ser de naturaleza matemática 2718281828459.

Averigua cómo se decide si un poema es matemático a  $n$ , y luego escribe una solución recursiva que permita decidir dado un  $n$  y un *poema*, si el poema es matemático de  $n$  o no

95. Una de las cosas más maravillosas de la naturaleza es la nieve, y en especial su partícula más pequeña que es el copo de nieve. Kotch ha planteado en su artículo “Acerca de una curva continua que no posee tangentes y obtenida por los métodos de la geometría elemental”, una manera interesante de describir un copo de nieve, que es una curva que se forma de tres partes iguales.

La forma de conseguir cada una de las partes es recurrente y considerando varios niveles de aplicación de la forma. En la figura 2.4, se muestra la forma que asume cada parte de la curva, considerando un nivel de profundidad:

Para nuestro problema, debes escribir una solución recursiva que dado un nivel de aplicación a la curva de Kotch me genere la cantidad de triángulos que se necesitan para generar una parte de la misma. En la figura 2.4 se muestra que si se tiene nivel 1 de aplicación se requiere 1 triángulo, si se quiere nivel 2 se necesitan 5, para nivel 3 se requieren de 21 y para nivel 4 de 85, y así sucesivamente.

96. En matemáticas hay muchas teorías y conjeturas interesantes, una de ellas es la sucesión alícuota de un número entero positivo es una descomposición del número hasta llegar a 0. Te mostramos a continuación la sucesión alícuota de 10: que es  $\{10, 5, 2, 1, 8, 4, 2, 1, 7, 1, 1, 0\}$

Lo que se hace para conseguir esto es ir reduciendo de la siguiente manera:

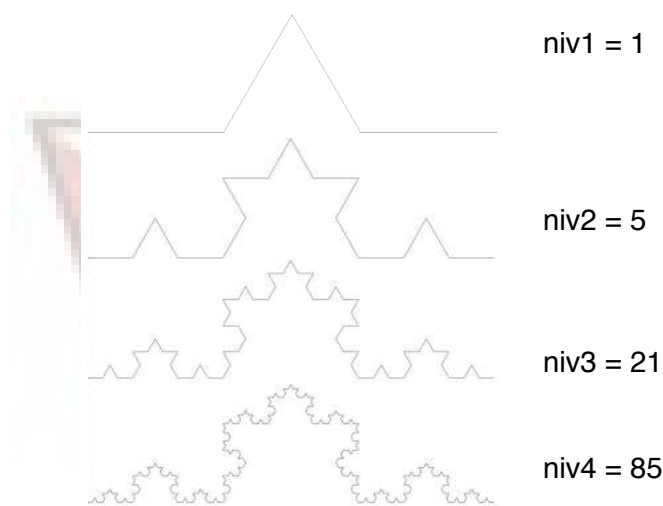


Figura 2.4: Niveles de aplicación de la curva continua de Kotch

$10 \rightarrow 10 + 5 + 2 + 1 - 10 = 8$   
 $8 \rightarrow 8 + 4 + 2 + 1 - 8 = 7$   
 $7 \rightarrow 7 + 1 - 7 = 1$   
 $1 \rightarrow 1 - 1 = 0$

Escribe el (los) método(s) RECURSIVOS necesarios para poder encontrar la sucesión alícuota de un número  $n$ .

97. <sup>2</sup> “La vida de un crítico es sencilla en muchos aspectos, arriesgamos poco, y tenemos poder sobre aquellos que ofrecen su trabajo y su servicio a nuestro juicio. Prosperamos con las críticas negativas, divertidas de escribir y leer- pero la triste verdad que debemos afrontar, es que en el gran orden de las cosas, cualquier basura tiene más significado que lo que deja ver nuestra crítica. Pero en ocasiones el crítico si se arriesga cada vez que descubre y defiende algo nuevo.. el mundo suele ser cruel con el nuevo talento, las nuevas creaciones.. lo nuevo, necesita amigos.. Lo único predecible de la vida es que es impredecible..”

En resumen equilibrar la vista de un crítico no es nada fácil. Para ello se ha encontrado una manera interesante de saber si una apreciación es equilibrada: se divide la crítica en dos porciones de palabras y se suman las letras de estas porciones si son iguales es equilibrada (no hay palabra de desequilibrio), caso contrario se pide encontrar la palabra del desequilibrio, que se busca siempre en la porción que haya dado mayor suma. Este trabajo se hace varias veces hasta ya no poder realizar la división.

Si se toma sólo una porción de la reseña:

cualquier basura tiene mas significado que lo que deja ver nuestra critica Pero

Se obtiene el siguiente resultado de las letras que tienen las palabras:

9, 6, 5, 3, 11, 3, 2, 3, 4, 3, 7, 7, 4

<sup>2</sup>Propuesto por Lic. Rossmery Torrico

La suma de la primera mitad es:  $9 + 6 + 5 + 3 + 11 + 3$  que es 37, y la suma de la segunda mitad es:  $2 + 3 + 4 + 3 + 7 + 7 + 4$  que es 30, entonces la palabra del desequilibrio esta en la primera mitad ya que su suma es mayor, si sigues haciendo esto la palabra del desequilibrio es “basura”.

Si se tiene, la siguiente reseña: “tengo yo la razon”, esa no tiene palabra de desequilibrio.

Para realizar este problema puedes asumir que los acentos se excluyen de la reseña, así mismo solo se tiene las letras del alfabeto inglés.

98. Hay un gusanito viajero y viejito, que quiere visitar una zona que esta conformada de varias lomas, sin embargo como comprenderás para él es difícil subir las cuevas empinadas por muy chicas que sean e incluso bajarlas, y ha decidido solamente visitar una loma de la zona. Por lo que se te ha pedido que le puedas decir cual es la loma que debería visitar de manera que no implique mucho esfuerzo por el tema de las pendientes, ya sea al subir o al bajar. En la Figura 4.4 se muestra un conjunto de lomititas a visitar, en la línea de abajo se tienen las lomas con datos discretos (secuencia de números) que es lo que tendrás como dato de entrada.

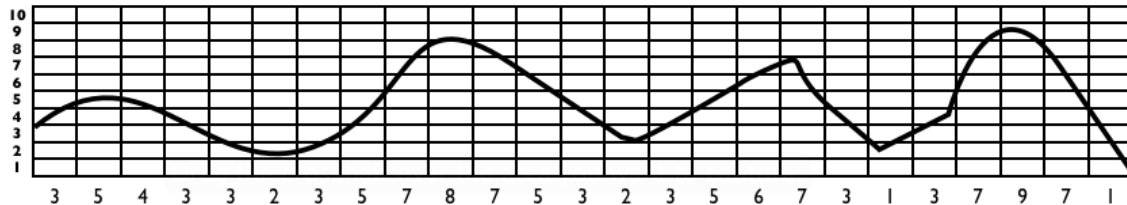


Figura 2.5: Las lomas que existen en la zona de visita del gusanito

Debes considerar que para el gusanito si se tiene una pendiente mayor o igual a  $p$ , esa ya no es una loma deseable para él. Debes indicar cual es la loma de mayor superficie que puede visitar y que no incumpla la cuestión de la pendiente.

La pendiente entre dos puntos se puede calcular como la diferencia entre ellos.

En definitiva para resolver el problema, te darán la secuencia de números que representan las lomas y la pendiente  $p$  que siempre será un número entero positivo.

Por ejemplo, si se considera la secuencia de la Figura 4.4 y pendiente 3, se tendría:

$sec = \{3, 5, 4, 3, 3, 2, 3, 5, 7, 8, 7, 5, 3, 2, 3, 5, 6, 7, 3, 1, 3, 7, 9, 7, 1\}$

$p = 3$

Entonces hay 4 posibles lomas de visita:

$loma1 = \{3, 5, 4, 3, 3, 2\}$

$loma2 = \{2, 3, 5, 7, 8, 7, 5, 3, 2\}$

$loma3 = \{2, 3, 5, 6, 7, 3, 1\}$

$loma4 = \{1, 3, 7, 9, 7, 1\}$

La loma1 y loma2 cumplen la condición de la pendiente, sin embargo la loma3 y la loma4 no, ya que al bajar de 7 a 3 hay una pendiente mayor a  $p$ , lo mismo en la loma4, subir de 3 a 7, o bajar de 7 a 1 son mayores que  $p$ .

Entre las dos lomas válidas la mejor es la loma2 porque cubre mayor superficie, por lo que el resultado es:  $\{2, 3, 5, 7, 8, 7, 5, 3, 2\}$ , en caso de no existir una loma con las condiciones del gusanito el resultado sería:  $\{\}$ .

Se te pide realizar el (los) método(s) recursivos necesarios para poder dar la loma que cumpla con las condiciones del gusanito.

99. Se tiene el siguiente problema problemático, de generar una cadena parsimoniosa y armoniosa que va así:

```
parmoniosa(1) = **\/**
parmoniosa(2) = ++****/++
parmoniosa(3) = **\++****/++/**
```

Se te pide encontrar la solución recursiva para *parmoniosa(n)*.

100. Dado un tablero de  $n \times m$ , en el que puede haber verduras (V) o nada (N), se pide indicar cuantas de las verduras un caballo puede comer si se asume que la forma de moverse de este caballo se asemeja a las de una pieza de ajedrez - es decir - se mueve en L. Para ello se te pide realizar los métodos **RECURSIVOS** necesarios para resolver el problema.

101. <sup>3</sup> Una tienda quiere optimizar el tiempo de atención con la ayuda de un programa que permita determinar cómo devolver cambios a sus clientes. Por ejemplo se tienen organizado el dinero de la siguiente forma: monedas de 1, 2 y 5 bolivianos, y billetes de 10, 20 y 50 Bs, para simplificar el problema los precios en la tienda no contienen centavos por lo que calcular un cambio siempre llevará a un monto sin centavos. El programa debe dado un monto obtener todas las posibles maneras de devolver el cambio.

Por ejemplo: si se tiene el monto 7 Bs, el resultado debería ser:

```
1 Bs: 1 moneda, 2 Bs: 3 monedas
1 Bs: 2 monedas, 5 Bs: 1 moneda
2 Bs: 1 moneda, 5 Bs: 1 moneda
```

Escribe el (los) método(s) RECURSIVOS necesarios para poder encontrar las diferentes maneras de devolver el cambio dado un monto  $m$ .

102. Mama Gallina llevó a sus pollitos a la fiesta de Urkupiña, y tuvo grandes complicaciones a la hora de alimentarlos, ya que no tuvo la previsión de comprar los asientos de manera que pueda pasar la comida entre los pollitos.. FUE TODO UN DRAMA!! Mama Gallina previsoría como es, ha decidido tomar sus recaudos para el próximo año y te pide que le ayudes a ubicar cual sería la mejor ubicación para ella y sus  $n$  pollitos, como sabes ya se aproxima el desfile de Cochabamba y quiere hacer una prueba de tu solución, ella tiene muñeca con las que venden las graderías y tiene la información de las filas y asientos libres y ocupados; y sobre la base de ello tiene que dar las opciones de donde puede comprar los asientos de manera que pueda pasar la comida entre los pollitos sin molestar a nadie.

Por ejemplo, si planea ir con 5 de sus pollitos y se tiene la siguiente configuración de la gradería (O = ocupado, L = libre) de 5 filas y 14 asientos por fila:

```

 1
 12345678901234
fila1 00000000000000
fila2 000000000000L0
fila3 LL000000LL00L0
fila4 000000LL00LL00
fila5 00000000000000
```

<sup>3</sup>Propuesto por Lic. Helder Fernandez

no existe un buen bloque de asientos que permita a Mami Gallina alimentar sin dificultad a sus pollitos :(, entonces probablemente no ira al desfile, en este caso le debes decir que “No existe asientos”.

Pero si consigue una graderia asi:

|       |                |
|-------|----------------|
|       | 1              |
|       | 12345678901234 |
| fila1 | 00000000000000 |
| fila2 | 000000LL0000LO |
| fila3 | LL0000LL0000LO |
| fila4 | 00000LL0000000 |
| fila5 | 00000000000000 |

Debería comprarse los asientos (fila2,7), (fila2,8), (fila3,7), (fila3,8), (fila4,6), (fila4,7); recuerda que tienes que considerar el asiento para Mama Gallina. Mama Gallina no quiere pasar la comida por encima de la cabeza de uno que no sea sus pollitos.

Ayuda a Mama Gallina y define los metodos necesarios RECURSIVOS para indicar que asientos deberia comparse, dado que se tiene el plano de la graderia y la cantidad ed pollitos que iran con Mami Gallina.

103. Se desea saber cuanto de tinta se gastara en mililitros para poder imprimir una flor inscrustada, se sabe que por cada centimetro se usa 1 mililitro; ademas se sabe que la flor mas pequeña esta incrustada en un cuadrado de lado 1 cm y para esta flor se requiere 10.28 ml, cuando se quiere hacer la flor de tamaño 2, esta incrustada en un cuadrado de lado 1.5 cm, es decir, el lado del nuevo cuadrado es siempre 50 % mas grande y la cantidad de tinta que se requiere es 25.70 ml, y asi sucesivamente. Para darte mas ayuda de lo que se esta hablando en la Figura 2.6 se te muestra su formación. (NOTA: perímetro de un circulo es  $2 * radio * \pi$ )

Dado este contexto se pide realizar el(los) método(s) RECURSIVOS necesarios para hallar la cantidad de tinta que se requiere para imprimir una flor incrustada de nivel  $n$ .

104. <sup>4</sup> “Los semánticos” es un grupo de jóvenes investigadores que estan estudiando reconocimiento semántico de documentos, para decidir de que se trata el documento y poder identificar palabras claves del mismo. Para ello han tomado una técnica sencillísima que es identificar todas aquellas palabras del documento que sean mayores a una longitud *longi* y contar su frecuencia de aparición. Por el momento ellos están concentrados en saber cuales son las palabras del documento que cumplen que su longitud es mayor a *longi*, y quieren obtener un reporte de las palabras que cumplen esta condición y NO quieren que el reporte contenga palabras repetidas - por el momento.

Por ejemplo si *longi* es 3 y se tiene el siguiente texto:

NO TE RINDAS AUN ESTAS A TIEMPO DE ALCANZAR Y COMENZAR DE NUEVO; ACEPTAR  
TUS SOMBRAS, ENTERRAR TUS MIEDOS, LIBERAR EL LASTRE, RETOMAR EL VUELO.  
NO TE RINDAS QUE LA VIDA ES ESO, CONTINUAR EL VIAJE, PERSEGUIR TUS SUEÑOS,  
DESTRABAR EL TIEMPO, CORRER LOS ESCOMBROS Y DESTAPAR EL CIELO; NO TE RINDAS  
POR FAVOR, NO CEDAS AUNQUE EL FRIO QUEME AUNQUE EL MIEDO MUERDA, AUNQUE EL  
SOL SE ESCONDA Y SE CALLE EL VIENTO AUN HAY FUEGO EN TU ALMA AUN HAY VIDA EN  
TUS SUEÑOS.

el reporte deberia ser:

<sup>4</sup>Propuesto por Lic. Rossmery Torrico

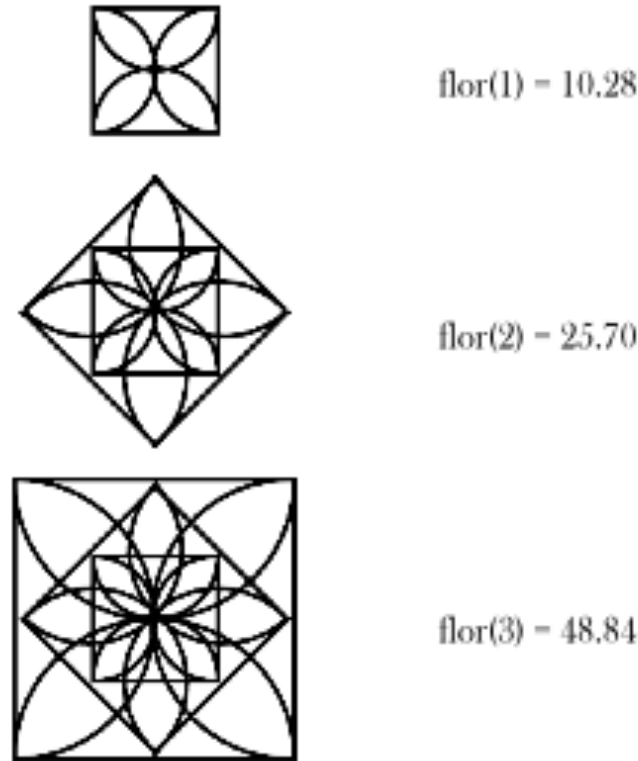


Figura 2.6: Flores incrustadas

{RINDAS, ESTAS, TIEMPO, ALCANZAR, COMENZAR, NUEVO, ACEPTAR, SOMBRAS, ENTERRAR, MIEDOS, LIBERAR, LASTRE, RETOMAR, VUELO, VIDA, CONTINUAR, VIAJE, PERSEGUIR, SUEÑOS, DESTRABAR, CORRER, ESCOMBROS, DESTAPAR, CIELO, FAVOR, CEDAS, AUNQUE, FRÍO, QUEME, MIEDO, MUERDA, ESCONDA, CALLE, VIENTO, FUEGO, ALMA}

Puedes asumir que el texto está escrito con el alfabeto en INGLÉS y que las palabras están en mayúsculas. Dado el contexto se te pide hacer el (los) método(s) recursivo(s) necesario(s) para resolver el problema.

105. Numerin es un chico muy activo en cuestiones de buscar retos matemáticos y se ha puesto la tarea de averiguar: cuál es la máxima suma que se puede conseguir de los valores positivos desde un punto inicio a un punto destino; dentro de un plano de números enteros. Numerin tiene prohibido pasar sobre los números negativos, sólo debe considerar los positivos, Numerin ama a pitágoras por lo que solamente camina en diagonal jamás de forma recta, él piensa que es mejor la hipotenusa que los catetos ;).

Diseña el (los) métodos recursivos necesarios para lograr el propósito de Numerin.

Por ejemplo si se desea llegar desde la posición inicial (0,0) a la posición final (3,5), en el siguiente plano:

```

1 -1 2 4 6 7 8
-3 1 6 7 -1 -1 -4

```



6 5 3 3 2 8 -5  
3 6 8 -4 6 7 1

El resultado será:  $1+1+6+6+3+7+2+7$  que permite obtener la máxima suma: 33. Obviamente una vez que considera un número en un camino no lo puede considerar de nuevo, es decir no puede volver a pasar por el número en un mismo camino.

106. Se aproxima la navidad y hay que armar el arbolito para ello necesitamos saber cuantas esferas de navidad se necesitan, lo cual dependerá del nivel del árbol que se quiera armar.

En la figura 2.7, se muestra la cantidad de esferas de acuerdo al nivel del árbol, diseña los procesos recursivos necesarios para poder encontrar el numero de esferas si se quiere armar un arbol de nivel n.

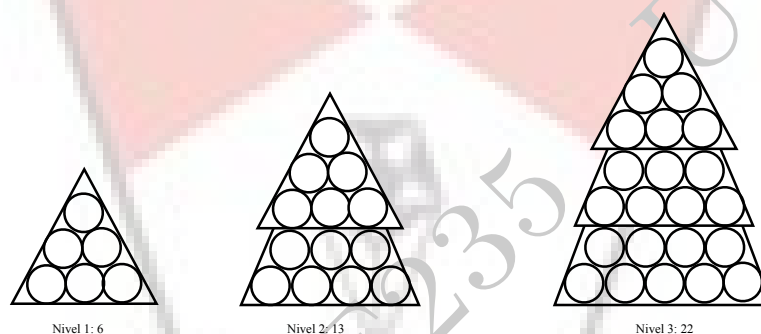


Figura 2.7: Arbol de navidad

107. Un numerito preocupado le dice a su papá, “somos muy aburridos los números, siempre valemos lo mismo”, su papá se ríe y le dice, “eso no es totalmente verdad, podemos mimetizarnos de muchas formas. Por ejemplo, tú hijo mio, que en valor eres un 8, puedes verte como:  $1+1+1+1+1+1+1$ ,  $1+1+1+1+1+2$ ,  $1+1+1+1+2+2$ ,  $1+1+2+2+2$ ,  $1+1+1+1+1+3$ ,  $1+1+3+3$ ,  $1+1+1+1+4$ ,  $1+1+1+2+3$ ,  $1+1+1+5$ ,  $1+1+2+4$ ,  $1+1+6$ ,  $1+3+4$ ,  $1+2+5$ ,  $1+7$ ,  $2+2+2+2$ ,  $2+2+4$ ,  $2+3+3$ , y así.. hay muchos tus :D”

El numerito se va muy alegre ya que su papá le ha demostrado que no es aburrido!!.

Dado un numero encuentra las diferentes formas en las que se puede mostrar, es bueno obviar los iguales, por ejemplo  $1+1+1+1+1+1+2$ , es igual a  $2+1+1+1+1+1$ . El orden no importa, lo que importa es encontrar todas las representaciones diferentes.

108. El mol de Hilbert es una curva rara que tiene un comportamiento de fractal como se muestra en la figura 2.8, lo que se quiere es calcular la cantidad de líneas que se requieren para imprimir el mol de Hilbert
109. En un país de las maravillas, se dice que hay muchas cosas maravillosas, entre ellas los viñedos, en los que se producen vinos de altura. De estos viñedos se tiene el registro de la producción por parcela; de cada parcela se tiene el tipo de uva y la cantidad, en el caso particular de este viñedo le interesa demasiado las uvas Moscatel de Alejandria las mismas que están identificadas por la letra M. Y las cantidades están expresadas en miles.

(M,70) (A,30) (V,20) (M,45) (M,34) (M,32) (M,43) (M,23) (M,10)  
(A,10) (M,34) (V,20) (N,65) (M,70) (N,10) (N,30) (M,45) (N,45)  
(N,10) (A,30) (V,20) (M,45) (M,34) (M,32) (M,43) (M,23) (M,10)  
(A,10) (A,34) (M,20) (N,65) (M,70) (N,10) (N,30) (M,65) (N,65)



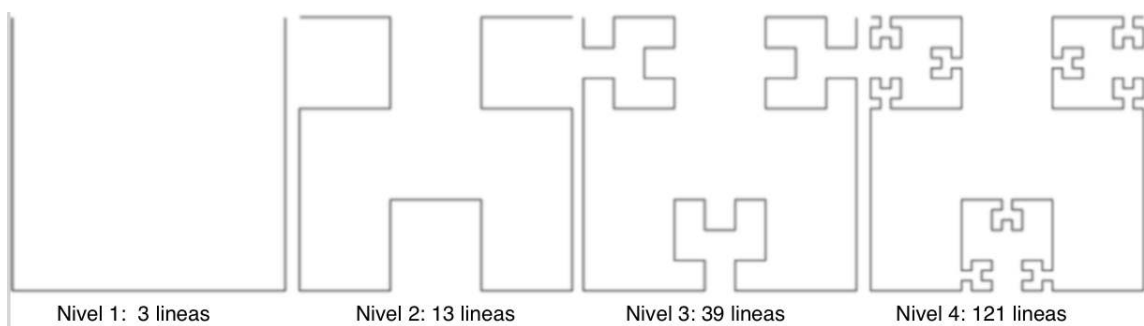


Figura 2.8: Mol de Hilbert

(A,70) (A,30) (V,20) (M,40) (N,34) (M,32) (M,43) (M,23) (M,20)  
 (N,10) (M,34) (M,20) (N,60) (M,90) (V,10) (N,30) (M,45) (N,59)

Se quiere encontrar la posición a partir de la cual se puede recoger la mayor cantidad de uva Moscatel de Alejandria, si se sabe que los cosechadores no pueden cosechar otra uva que no es la indicada, y sólo pueden manejar el coche de cosecha en línea recta. Solo se puede mover arriba, abajo, derecha o a la izquierda.

110. En el **Gran Hotel**, hay demasiados secretos, tantos que se necesita guardar muchos de ellos en cajas de seguridad que tienen una clave de acceso. Las cajas de seguridad, se abren sólo si se hacen girar los rodillos de la clave en las direcciones correctas y además para asegurar la imposibilidad de hackeo, con la mínima cantidad de giros..

Se necesita saber cual será la manera correcta de abrir las cajas. Por ejemplo si la clave fuera 456392000137458637, al cerrar la caja los cilindros se establecen con puros 0s, en nuestro ejemplo será así: 000000000000000000.

Al terminar se debe emitir el siguiente reporte: (4 arriba)(5 arriba)(4 abajo)(3 arriba)(1 abajo)(2 arriba)(0 arriba)(0 arriba)(0 arriba)(1 arriba)(3 arriba)(3 abajo)(4 arriba)(5 arriba)(2 abajo)(4 abajo)(3 arriba)(3 abajo).

Debes considerar que la clave puede ser de muchos dígitos, inclusive 100

111. <sup>5</sup> La clasificación es una técnica que “aprende” automáticamente cómo clasificar objetos. Este aprendizaje se basa en datos previamente etiquetados (clasificados), el proceso de “etiquetar” automáticamente tiene un alto costo en tiempo y esfuerzo. Todos los días se publican noticias en Twitter, muchas de ellas son similares. Se tienen diversos retos como generar una lista de palabras clave, que permitan agrupar dichas noticias de forma automática. En este sentido para formar parte del equipo de desarrollo tienes que probar tus destrezas de programación por lo que tienes la misión de aportar al Módulo de clasificación automática, implementando métodos recursivos que: devuelva la cantidad de titulares que contienen la palabra clave que se está procesando, con esta información se tomará la decisión si vale o no la pena crear un grupo con el tema que describe la palabra clave.

Se crea un grupo nuevo si el tema descrito por la palabra clave supera el 30 % del total de titulares publicados. Por ejemplo: Tenemos un texto que contiene 6 titulares de tweets publicados recientemente. Los mismos están separados por \*:

Los grandes países amenazan con unirse para aislar a Trump\* La victoria de Trump reaviva temores escondidos\* Borrachos a los 13 años\* Donald Trump será destituido, dice el profesor

<sup>5</sup>Propuesto por Lic. Rossmery Torrico

que predijo su triunfo electoral\* Obama se despidió del mundo con una defensa apasionada de la inmigración\* El 'muro' que Trump no se esperaba

Si la palabra clave es "Trump", existen 4 de 6 titulares que lo contiene eso equivale 66 % porcentaje que es mayor al 30 % por tanto SI se debe generar un grupo nuevo con el tema "Trump". Si la palabra clave fuese "Obama", existe 1 de 6 titulares, es decir 16 % porcentaje que es menor al 30 % por lo tanto NO se debe generar grupo nuevo con el tema "Obama".

112. Piensa en el juego de "Las damas", en ese juego tu puedes "comer" fichas enemigas saltando sobre ellas, asume que tienes fichas Negras (N), fichas Blancas (B) y espacios Vacíos (.). Por ejemplo si tienes una Negra una Blanca y un espacio Vacío - NB. - la negra puede saltar sobre la Blanca ("comersela") dejando la configuración así espacio Vacío, espacio Vacío y Negra - ..N.

Imagina el tablero como una grilla de valores N, B y .; entonces se puede pasar de la situación 1 a la 2 moviendo la ficha N de la posición 0,2 a la posición 0,4 "saltando" sobre la blanca; así se ha "comido" la ficha blanca; de la misma manera la situación 3 a la situación 4, muestra que la ficha N de la posición 1,3, puede saltar sobre la ficha B y ubicarse a la posición 3,1, en consecuencia se gana una ficha blanca más.

a). situación 1      b). situación 2      c). situación 3      d). situación 4

NBNB.NNB  
..NN.B.N  
N.BN.NBN  
B..N.N.N  
N..BNBNN  
...N.N.B  
...BNNBN

NB..NNNB  
..NN.B.N  
N.BN.NBN  
B..N.N.N  
N..BNBNN  
...N.N.B  
...BNNBN

NB..NNNB  
..NN.B.N  
N.BN.NBN  
B..N.N.N  
N..BNBNN  
...N.N.B  
...BNNBN

NB..NNNB  
..N..B.N  
N..N.NBN  
BN.N.N.N  
N..BNBNN  
...N.N.B  
...BNNBN

Realiza el (los) procesos recursivos que permitan de forma automática indicar cuantas fichas Blancas Máximo se han comido siguiendo las reglas del juego, en esta versión es posible saltar en todas las direcciones.

113. Se dice que un número es polidivisible si:

- Es mayor que cero.
- El número formado por su dígito más significativo es múltiplo de 1 (esto lo cumplen todos los números).
- El número formado por sus dos dígitos más significativos es múltiplo de 2.
- El número formado por sus tres dígitos más significativos es múltiplo de 3.
- El número formado por sus cuatro dígitos más significativos es múltiplo de 4.
- .. y así sucesivamente.

En general, el número formado por sus  $i$  dígitos más significativos es múltiplo de  $i$ .

Por ejemplo, el número 2016 es polidivisible de 4 dígitos, pues es mayor que cero y 2 es divisible por 1, 20 es divisible por 2, 201 es divisible por 3 y, por último, el propio 2016 es divisible por 4.

Sin embargo, el número 2225 no es polidivisible pues a pesar de que el 2 es divisible por 1, el 22 lo es por 2 y el 222 por 3, el propio 2225 no es divisible por 4.

Se quiere saber cuáles son los números polidivisibles de  $k$  dígitos.

Por ejemplo, si  $k$  es 3, la respuesta sería: {120, 180, 240, 300, 360, 420, 480, 540, 600, 660, 720, 780, 840, 900, 960}

Implementa el (los) métodos necesarios recursivos para encontrar a los números polidivisibles de  $k$  dígitos.

114. Mirar una estrella en su plenitud es hermoso, si se pudiera ver a detalle tendríamos estrellas estrelladas, pues bien, el trabajo es determinar la cantidad de líneas rectas que se necesitan para dibujar una estrella estrellada de nivel  $n$ . En la figura 2.9, se muestra el número de líneas que se requiere para formar una estrella de nivel 1, de nivel 2 y de nivel 3. Se te pide calcular el número de líneas rectas que se requieren para formar una estrella estrellada de nivel  $n$ .



Figura 2.9: Número de líneas que se utiliza para estrellas estrelladas

115. Los caballos en el ajedrez son fichas que se mueven de manera peculiar, si se tiene un caballo ubicado en la celda  $(x, y)$  del tablero se quiere saber todas las posiciones a las cuales puede moverse el caballo. (ver figura 2.10) Debes cuidar de no repetir las posiciones.

Realiza el (los) métodos necesarios recursivos para determinar la lista de TODAS las posiciones (no repetidas) a las que puede llegar un caballo haciendo uso de sus movimientos característicos.

116. Nosila y Anairda son muy inteligentes y les gusta los retos. Han encontrado unas secuencias de números; y se han puesto a la tarea de averiguar cuántas subsecuencias  $\{a_l, a_{l+1}, a_{l+2}, \dots, a_r\}$  cuando  $1 \leq l \leq r \leq n$ , que cumplen que todos sus elementos tienen una diferencia menor igual a  $k$ .

Por ejemplo:

$\{3, 5, 7, 1, 2, 4, 5, 7, 2\}$

y  $k$  es 3, ellas han encontrado que el resultado es 13, ya que las subsecuencias que siguen a continuación cumplen con la condición:

$\{3\}, \{1\}, \{2\}, \{2\}, \{3,5\}, \{5,7\}, \{1,2\}, \{1,2,4\}, \{2,4\}, \{2,4,5\}, \{4,5\}, \{4,5,7\}, \{5,7\}$

Implementa el (los) método(s) **RECURSIVOS** necesario(s) para resolver el problema.

117. *RTB* Indique de forma detallada cuál es la secuencia numérica generada por el método recursivo  $f()$  en el listado siguiente si la llamada es  $f(1), f(2), f(3), f(4) \dots f(10)$ .

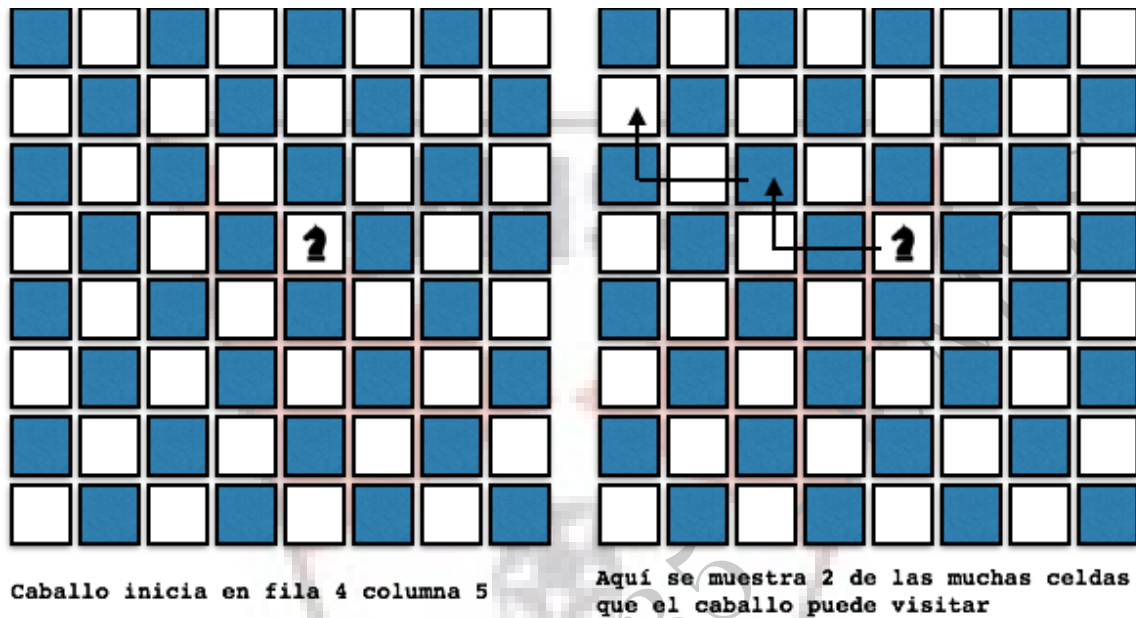


Figura 2.10: Ejemplo del problema

```
public long f(int n){
 long num;
 if(n == 0 || n == 1)
 num = 1;
 else if(n % 2 == 0)
 num = 2 + f(n - 1);
 else
 num = 3 + f(n - 2);
 return num;
}
```

118. *HF* Se dispone de un tablero de 8 casillas similar al que se muestra a continuación:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| X | O | X | X | O | O | O | X |
|---|---|---|---|---|---|---|---|

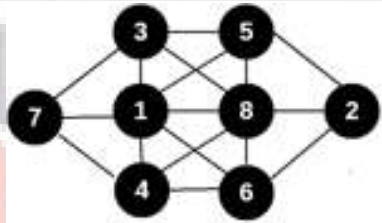
En la situación inicial, las casillas están ocupadas por cuatro *O* y cuatro *X* distribuidas de forma aleatoria. En cada paso la única operación posible es intercambiar dos fichas adyacentes. Programar de la forma más eficiente posible el proceso que devuelva la secuencia de intercambios para ordenar el tablero de forma que todas las fichas tengan al menos otra igual adyacente.

El ejemplo que se dio la secuencia óptima de movimientos, considerando las celdas del tablero desde la posición 1 es: intercambiar 2 con 3, intercambiar 4 con 5, intercambiar 5 con 6, intercambiar 6 con 7.

Tu solución **DEBE SER RECURSIVA**

119. *HF* Escribir un programa que busque la forma de rellenar 8 números en una cuadrícula con las condiciones dadas. Se solicita colocar los números 1, 2, 3, 4, 5, 6, 7, 8 en los ocho círculos en la figura que se muestra en la figura 119, de tal manera que ningún número sea adyacente a un número que esté al lado de

la secuencia. Por ejemplo, 1 no debe ser adyacente a 2, pero puede ser adyacente a 3, 4, 5, 6, 7, 8. De manera similar para otros. Lo que se ve en la figura es un ejemplo de ubicación, realiza el(los) proceso(s) necesario(s) para indicar todas las posibles soluciones al problema



Puedes imaginar la imagen de la siguiente manera:

```

. V V .
V V V V
. V V .

```

Donde los **V** deben ser reemplazados por los valores del 1 al 8.

120. *MLBC* Considera los números de la siguiente secuencia:

```

1 1
2 12
3 123
4 1234
5 12345
.
.
10 12345678910
.....

```

Cada uno de ellos está en una posición, por ejemplo el número de secuencia de la posición 5, es 12345.

Lo que se pide es dado una posición **n**, indique cuantos números de la secuencia es divisible entre 3.

Por ejemplo, si **n** es **7**, la respuesta es **4**, ya que cuatro números de la serie del **1** al **7** son divisibles entre **3**.

Realiza el (los) métodos necesarios recursivos para determinar la cantidad de números de los **n** números de la secuencia son divisibles entre 3. Cuida tus precondiciones para tu solución.

## Conceptuales

1. Indica que aseveraciones son verdaderas y porque.

a) Se tiene la interfaz: *Rastreable*

```

public interface Rastreable {
 int calcularPosicion();
 String direccion(int p);
}

```

¿Cuál pedazo de código compilaría? (Seleccionar todas las que se aplican)

- 1) `interface Loalizable implements Rastreable {}`
- 2) `abstract class Misil extends Rastreable {}`
- 3) `abstract class Avion implements Rastreable {}`
- 4) `abstract class Nave implements Rastreable {  
 public int calcularPosicion() { return 0;}}`
- 5) `class Class2 implements Rastreable {  
 int calcularPosicion( ) { return 0; }  
 String direccion(int p) { return "Posicion " + p; }}`

b) Si se tiene la siguiente jerarquía de clases:

```
public interface A {
 int metA1();
 String metA2();
}
public class B implements A{
 public int metA1(){...}
 public String metA2(){...}
 public double metB1(){...}
}
```

¿Cómo ejecutaría cada pedazo de código? Asume condiciones ideales para cada ejecución.

- 1) `A obj = new B();`
- 2) `B obj1 = new B();`
- 3) `obj instanceof B`
- 4) `obj1 = (B)obj;`
- 5) `obj.metA1();`
- 6) `obj1.metB1();`

## Capítulo 3

# Estructuras de Datos Lineales

Este capítulo tiene ejercicios extraídos de exámenes de la materia y algunos son adaptados de ejercicios de la ACM ICPC.

1. Dada una Lista de Simple Enlace:

- Contar la cantidad de elementos que tiene la lista.
- Aplicar un **Functor** que reduzca los elementos de la lista. Todo **Functor**, sabe reducir de dos modos: 1) Si recibe dos elementos devuelve uno y 2) Si recibe uno devuelve otro. Por ejemplo sea **f** un Functor, entonces se puede enviar los siguientes mensajes a **f**: **f.reduceir(elem1, elem2)** o **f.reduceir(elem)**. En ambos casos el resultado es un elemento.
- Invertir los enlaces de la Lista.
- Concatenarla con otra lista de simple enlace. Debes tener cuidado de no tener elementos repetidos en la lista resultante.
- Dividirla en dos listas de longitudes lo más similares posible. Asume que no puedes usar el método contar elementos.
- Que contiene elementos **Comparables**, realiza los métodos que permiten insertar los elementos de forma ordenada ascendentemente.

2. Se desea modelar una solución para resolver el problema de atención al cliente en una entidad pública. Se sabe que los clientes vienen a la institución por distintos motivos, básicamente por 5 tipos de quejas:  $q1, q2, q3, q4$  y  $q5$ . Para ello se ha dispuesto 3 ventanillas de atención: *vent1*, *vent2* y *vent3*. Las ventanillas *vent1* y *vent3* se especializan en atender quejas del tipo  $q2$  y  $q5$ , en caso de emergencias y cuando existe demanda pueden atender quejas  $q3$ . La *vent2* atiende las quejas  $q1, q3$  y  $q4$ . La única vez que la ventanilla *vent1* o *vent3* atienden una queja  $q3$ , es cuando la *vent2* solicita ayuda. En este caso se da la prioridad de atención al cliente con queja  $q3$  en ya sea la ventanilla *vent1* o *vent3*.

- Describe que estructura(s) de datos se adecúa mejor para resolver el problema.
- Describe la estructura de los elementos que se almacenarán en la(s) estructura(s) de datos que elegiste.
- Implementa la funcionalidad necesaria para permitir la llegada de los clientes a la entidad.
- Implementa la funcionalidad necesaria para permitir la atención de los clientes.
- Si además el gerente de la entidad requiere saber cuál ha sido la productividad de cada una de las ventanillas, indica que cambios realizarías a tu anterior solución.



3. Dada una pila
  - a) Invierte los elementos de la Pila
  - b) Busca si existe un elemento
  - c) Se quiere saber cual es el elemento que está en la base de la Pila
  - d) Imagina un modo que permita migrar los elementos de la pila en una cola, de modo que el tiempo en el que ingresaron los elementos en la pila sea lo que rija el orden de inserción en la cola.
4. Se tiene una mesa redonda tradicional para compartir comida china. Luis es uno de los comensales que comparten de la comida; se desea saber en instantes específicos de tiempo qué plato tiene delante de él. Dado que los platos se encuentran en una plataforma que gira en el centro de la mesa, lo único que se sabe es que se tienen  $n$  comensales, y cada uno de ellos ha pedido una variedad distinta de plato. Entonces uno de los comensales puede servirse de alguno de los platos, para ello debe girar la plataforma central hasta tener el plato de su elección al frente. En ese instante de tiempo, obviamente es probable que Luis tiene delante de él algún plato particular, es en éste momento que el programa debería sacar un reporte de qué plato tiene en frente Luis.
  - a) Describe que estructura(s) de datos se adecúa mejor para resolver el problema.
  - b) Describe la estructura de los elementos que se almacenaran en la(s) estructura(s) de datos que elegiste.
  - c) Implementa la funcionalidad necesaria para permitir la petición de un plato específico por uno de los comensales.
  - d) Implementa la funcionalidad necesaria para saber qué plato de comida tiene Luis al frente.
5. Sean A, B y C conjuntos de elementos, se desea:
  - a) Realizar la union de A y B en C
  - b) Realizar la intersección de A y B en C
  - c) Realizar la diferencia de A y B en C
6. Dados dos polinomios, se desea:
  - a) Sumarlos
  - b) Multiplicarlos
7. Se desea saber si un mapa se puede colorear con dos colores, dado que se tiene la información de las conexiones entre las regiones del mapa. Las conexiones son duplas que se representan de la siguiente forma:  $(reg1, reg2)$ . Cada región tiene nombre y color de pintado (si estuviera pintado). Tu trabajo es decidir si el mapa puede ser o no pintado de dos colores.
8. En un gran restaurante de tres estrellas “michelin”, se sirven desayunos y como es de suponer, la presentación es un detalle que no debe descuidarse a tiempo de servir todas las comidas.

Una de las opciones de desayuno que tiene es panqueques franceses, el chef elabora un pedido de  $n$  panqueques y va poniéndolos en un platillo a medida que los cocina, al finalizar el chef debería entregar el plato con los  $n$  panqueques puestos de forma que el panqueque de mayor diámetro vaya siempre abajo y el de menor arriba. Ayuda a chef para saber cuántos movimientos se necesitan hacer para ponerlos en el orden estipulado, recuerda que los panqueques están uno sobre otro, por lo que para cambiarlos de lugar probablemente necesites voltear una pila de panqueques.



9. Se desea tener una aplicación que permita registrar las solicitudes de una tienda que sólo atiende a clientes vía correo electrónico. Los pedidos que llegan deben tener un cliente, dirección - zona y una lista de objetos que se pide. Cada objeto debería tener un peso que defina su posibilidad de atención y en base a la suma de los pesos de los objetos de los pedidos se define el orden de atención de estos pedidos. Los pedidos jamás pesan 100 ptos. y lo mínimo que pesan es 10 ptos. las solicitudes se organizan de acuerdo al peso que tienen, del 10 al 19, del 20 al 29, y así sucesivamente del 90 al 99. Cada vez que se atiende el pedido, se debe a su vez registrar el pedido de acuerdo a la dirección del cliente. Al finalizar el día, la tienda, saca un listado que tiene los pedidos organizados por dirección - zona, cosa que al día siguiente su personal de reparto se haga cargo de la entrega de un conjunto de pedidos, se sabe que el personal de reparto se divide el trabajo por zonas
- Elige la(s) estructura(s) de datos más adecuada(s) para representar el problema.
  - Describe la estructura de los elementos que viven en la estructura de datos elegida.
  - Escribe el (los) método(s) necesario(s) para poder atender el requerimiento de un cliente y registrarlo para su posterior entrega. Debes cuidar que la estructura que elijas debe ser lo más eficiente posible.
  - Escribe el (los) método(s) necesario(s) para poder encontrar la lista que permita planificar las entregas cada día.
10. *Lic. R. Salazar* Dada una cadena se pide encontrar su cadena extendida. Esta cadena se extiende por las vocales fuertes que tiene (se llaman vocales fuertes a la **a, e y o**), cada vez que aparece una vocal fuerte en la cadena a ésta se debe anteponer con la vocal débil más cercana y posponer la vocal débil mas lejana a ella. Por ejemplo: la cadena **extiende** se transforma en **ieuxtieundieu**, la única regla es que si ya una de las vocales débiles la antepone o pospone (en el orden) ya no debería considerarse, en el ejemplo, ante la segunda aparición de la **e** en **extiende**, ya esta antepuesta con **i** por lo tanto ya no es necesario ponerle de nuevo la **i**. Otro ejemplo es la palabra **cuando** que se transforma en **cuiaundiou**, en este caso la primera vocal débil mas cercana es **i** no **u**, por lo que se antepone la **i** a la **a** y se pospone la **u**.
- Elige la(s) estructura(s) de datos **MÁS ADECUADA(S)** para representar el problema.
  - Describe la estructura de los elementos que viven en la estructura de datos elegida.
  - Escribe el (los) método(s) necesario(s) para dada una cadena encontrar la extendida.
11. En una fiesta de colores se ha decidido, invitar a personas que tienen que tener una pareja. Cada persona invitada, “elige” a su pareja y se supone que ésta pareja debería venir con el mismo color de ropa que la persona que la eligió. Por supuesto, en esta tarea de elegir parejas es posible que una persona necesite vestirse de mas de un color - lo cual no es permitido.
- Dada una secuencia de personas invitadas y sus “elegidas”, indica cuáles son las parejas posibles, e indica cuáles personas **NO PUEDEN** ir a la fiesta - que son aquellas que se les pide vestirse de más de un color.
- Elige la estructura de datos mas adecuada para resolver el problema
  - Describe la estructura de los elementos que se almacenarán en tu estructura de datos
  - Escribe el (los) método(s) necesario(s) para indicar las parejas válidas y las personas que quedan fuera de la fiesta.
12. La embotelladora **bolem** necesita un programa que le ayude a distribuir sus productos estrella: **Ko-ka&Kolla y Fantas-tiK** por la ciudad. Para ello acude a la UMSS en busca de personas que le solucionen el problema. Se decide que la modalidad de elección de los estudiantes que serán contratados para resolver el problema, es pidiendo a los estudiantes puedan resolver el siguiente problema más sencillo.
- La empresa tiene un camión que debe repartir sus productos en la ciudad a sus distribuidores. Los pedidos de productos se anotan en una cola. Para cada pedido se anota la cantidad de unidades y el distribuidor. Un

distribuidor puede ser mayorista o minorista. El mayorista siempre pide el producto por cajas. El minorista las pide por cantidad de botellas. De cualquiera de ellos, **bolem** anota el nombre del distribuidor (que se denomina razón social) y su dirección. Estos pedidos deben ser puestos en la cola de espera por prioridad: los mayoristas son prioritarios a los minoritarios.

Para realizar el despacho se procede como sigue: hay que llenar el camión con cajas de bebidas. El camión puede llevar hasta 20 cajas de bebidas (cada caja tiene 12 botellas), aunque la empresa está pensando en la posibilidad de comprar un camión con mayor capacidad. El camión se llena con tantos pedidos como sea posible en el orden en que fueron catalogados para su atención. Es posible que un pedido no se pueda satisfacer totalmente, ya que el camión no tiene la suficiente capacidad como para atenderlo de una sola vez, entonces se carga el camión con tantas cajas como sea posible, dejando anotada la cantidad que queda pendiente a ser despachada.

En base a esta descripción se debe:

- Elegir las estructuras de datos más **ADECUADAS** que permitan representar los pedidos y el camión de la empresa.
- Realiza un modelo que muestre las estructuras elegidas, la empresa y los elementos de las estructuras.
- Escribir el código en Java de los métodos necesarios para realizar una carga del camión a partir de los pedidos.

**Caso Prueba 1** Un ejemplo para el inciso c) es:

-----  
\*\*\* LISTA DE PEDIDOS POR ORDEN DE LLEGADA\*\*\*

Los Molinos c. flores 123, total pedido: 10, quedan: 10, unidad: botellas  
Don Pele av. asia 4321, total pedido: 10, quedan: 10, unidad: cajas  
Los Molinos c. flores 123, total pedido: 10, quedan: 10, unidad: botellas  
Salteneria Tucuman av. argentina 10, total pedido: 50, quedan: 50, unidad: botellas  
Supermercado Lo Caro c. francia 440, total pedido: 30, quedan: 30, unidad: cajas

PEDIDOS ALMACENADOS POR PRIORIDAD

Don Pele av. asia 4321, total pedido: 10, quedan: 10, unidad: cajas  
Supermercado Lo Caro c. francia 440, total pedido: 30, quedan: 30, unidad: cajas  
Los Molinos c. flores 123, total pedido: 10, quedan: 10, unidad: botellas  
Los Molinos c. flores 123, total pedido: 10, quedan: 10, unidad: botellas  
Salteneria Tucuman av. argentina 10, total pedido: 50, quedan: 50, unidad: botellas

PRIMERA SALIDA DEL CAMION

Cargando 10 cajas.

Cargando 10 cajas. Quedan 20 cajas pendientes al pedido 2

SEGUNDA SALIDA DEL CAMION

Cargando 20 cajas. Quedan pedidos pendientes

TERCERA SALIDA DEL CAMION

Cargando 1 cajas.

Cargando 1 cajas.

Cargando 5 cajas. No quedan pedidos pendientes  
-----

13. **Fisgon**, ha tratado de capturar lo que se transmite en la red muchas veces, pero siempre ha fallado, debido a que **Vac** siempre encuentra alguna forma de evitar que esto suceda. **Vac** lo que hace es cifrar los mensajes que se transmiten y así evitar que sean vistos por **Fisgon**. Se conoce que **Vac** cifra los mensajes por el siguiente método:

- El borra todos los espacios y marcas de puntuación del mensaje original
- El reemplaza todas las letras sucesivas idénticas por una instancia de la letra
- El inserta pares de letras idénticas en distintos lugares muchas veces y de forma aleatoria.

La tarea que se debe hacer es el proceso inverso, es decir hay que tratar de restaurar un mensaje cifrado. Para esto, hay eliminar todos los pares de letras idénticas insertadas en el tercer paso del procedimiento. Como entrada se tiene un texto que contiene un mensaje cifrado por **Vac**. El mensaje consiste de puras letras en minúsculas.

La salida es el mensaje restaurado

**Ejemplo:**

| Entrada                               | Salida    |
|---------------------------------------|-----------|
| wwstdaadierfflitzzz                   | stierlitz |
| mcffceoorppnrnioirsaaooazrrzjooeqnnln | mensaje   |
| eeexafaafmccessndd                    | examen    |

- Define la estructura de datos adecuada para representar el problema.
  - Define la estructura de los elementos de tu estructura de datos.
  - Escribe el (los) método(s) necesarios para dado un mensaje cifrado, me permita restaurar y obtener el mensaje restaurado.
14. Una empresa de gestión de campañas, se encarga de hacer entre muchas cosas, la elaboración de banderines multicolores, los mismos que deben cumplir que jamás en la tira resultante de banderines se tenga dos consecutivos del mismo color. Para ello, disponen de una secuencia de banderines de los colores de los que se desea construir la tira de banderines. Esta secuencia se da en forma desordenada y aleatoria en colores, jamás en esta secuencia de entrada se tienen más de 3 banderines de un mismo color seguidos.

Por ejemplo: si se tiene la siguiente entrada de banderines de color, donde *r* es rojo, *v* es violeta y *a* es anaranjado:

arrvrvrvrarvaaarvva

La salida correcta que permita tener una tira de banderines tricolor mezclado seria:

arvrrrvrvrarvaravava

En la entrada, se tiene la misma cantidad de banderines de distintos colores, en el ejemplo de cada color se tiene 6 banderines, por lo que, siempre es posible tener una tira de banderines que cumplan las condiciones especificadas.

Por otro lado, si se tiene la siguiente secuencia:

arrvrvrrrvaaaraavv

La salida correcta que permita tener una tira de banderines tricolor mezclado seria:

aravrrrvrvrvaravava

- a) Escoge la estructura de datos adecuada para resolver el problema.
  - b) Describe la estructura de los elementos que almacenarás en las estructuras que elegiste en el inciso a.
  - c) Escribe el (los) método(s) necesario(s) para poder dar una secuencia de entrada de banderines consiguiendo la tira de banderines correcta.
  - d) Imagina que no existe la condición de que en la entrada se tiene la misma cantidad de banderines por color, ¿tu solución aún funciona? ¿Por qué?
15. Los navegadores web estándares contienen características para moverse hacia atrás y hacia adelante entre las páginas recientemente visitadas. Una manera de implementar estas características es usar dos pilas para guardar el rastro de las páginas, que pueden ser alcanzadas moviéndose hacia atrás o hacia adelante. Los navegadores cuentan con los siguientes comandos:

**BACK:** Poner la página actual en el cima de la pila **adelante**. Retirar la página del tope de la pila **atras**, haciendo de ella la nueva página actual. Si la pila **atras** está vacía, el comando es ignorado.

**FORWARD:** Poner la página actual en el tope de la pila **atras**. Retirar la página del tope de la pila **adelante**, haciendo de ella la nueva página actual. Si la pila **adelante** es vacía, el comando es ignorado.

**VISIT < url >:** Poner la página actual en el tope de la pila **atras**, y hacer que el URL especificado sea la nueva página actual. La pila **adelante** es vaciada.

**QUIT:** Permite salir del navegador.

Asuma que el navegador inicialmente carga la página web en la URL **http://www.acm.org/**

A continuación se muestra un ejemplo de cómo debería comportarse el navegador si se tiene la siguiente secuencia de ejecución:

```
VISIT http://acm.ashland.edu/
VISIT http://acm.baylor.edu/acmicpc/
BACK
BACK
BACK
FORWARD
VISIT http://www.ibm.com/
BACK
BACK
FORWARD
FORWARD
FORWARD
QUIT
```

La salida seria:

```
http://acm.ashland.edu/
http://acm.baylor.edu/acmicpc/
http://acm.ashland.edu/
http://www.acm.org/
Ignored
http://acm.ashland.edu/
```

```
http://www.ibm.com/
http://acm.ashland.edu/
http://www.acm.org/
http://acm.ashland.edu/
http://www.ibm.com/
Ignored
```

Como podrás notar, las palabras: **VISIT, BACK, FORWARD, QUIT**, indican las diferentes acciones que se pueden hacer en el navegador. Y cada línea de la salida responde a cada una de las instrucciones. Cuando no es posible, ya sea ir adelante o atrás se debe generar un mensaje **Ignored**.

- a) Escribe el (los) método(s) necesario(s) para poder responder la acción **VISIT**.
- b) Escribe el (los) método(s) necesario(s) para poder responder la acción **BACK**.
- c) Escribe el (los) método(s) necesario(s) para poder responder la acción **FORWARD**.
- d) Escribe el (los) método(s) necesario(s) para poder responder la acción **QUIT**.

16. Lulú una niña muy inquieta, para navidad quiere regalarle un collar a su mejor amiga, pero no quiere que sea uno más del montón por lo que ha decidido comprar piedras de distinto color y hacer un collar colorido. Pero ella jamás quiere formar el collar juntando piedras de color morado y negro, naranja y negro, ni juntar dos piedras del mismo color, pues le parece de mal gusto. Debes ayudar a Lulú a indicar si la cantidad y variedad de piedras que ha comprado le servirán para hacer un collar como ella quiere y cual seria la disposición de las piedras. En caso de no poder hacer el collar debes decir “No es posible”.

Por ejemplo:

Caso 1:

ENTRADA:

{morado, negro, verde, morado, verde, negro, negro, morado}

SALIDA:

No es posible

Caso 2:

ENTRADA:

{morado, negro, amarillo, verde, naranja, negro, rojo, blanco, amarillo}

SALIDA:

{morado, amarillo, negro, verde, naranja, rojo, negro, blanco, amarillo}

Caso 3:

ENTRADA:

{morado, negro, negro, morado, verde, verde, amarillo, naranja, blanco, rojo}

SALIDA:

{morado, verde, negro, verde, negro, amarillo, morado, naranja, blanco, rojo}

- a) Elige la(s) estructura(s) de datos **MÁS ADECUADAS** para representar el problema.
  - b) Describe la estructura de los elementos que viven en la estructura de datos elegida.
  - c) Escribe el (los) método(s) necesario(s) para dada la lista de piedras que tiene Lulú, intentar generar un collar de acuerdo a las necesidades y condiciones de Lulú.
17. Jolly Jumpers, es una notación que se asigna a una serie de números cuya distancia permite realizar saltos mas pequeños que la cantidad de números que tiene la secuencia. Por ejemplo: 3, 4, 5, 7, 2, 3 es un Jolly Jumper, ya que la cantidad de números que tiene es 6 y los saltos que debe hacer entre número y número

siempre son menores que 6; del 3 al 4 salto de 1, del 4 al 5 salto de 1, del 5 al 7 salto de 2, del 7 al 2 salto de 5 y así sucesivamente.

Realiza el programa que dado una serie de números enteros positivos, me permita decidir si es un Jolly Jumper o no y además indicar el (los) lugares donde se ha generado ruptura del Jolly Jumper. Es de más decir que una secuencia de 1 elemento es Jolly Jumper!!!

Sobre la base de esta descripción se pide:

- Identificar las estructuras de datos MAS adecuadas para modelar el dominio del problema.
- Definir de forma clara la estructura de los elementos que contendrán las estructuras de datos antes elegidas
- Implementar el (los) método(s) necesario(s) para dada una secuencia de números decidir si es o no Jolly Jumper y además registrar las rupturas en la secuencia si existiesen

Por ejemplo:

| ENTRADA            | SALIDA                              |
|--------------------|-------------------------------------|
| 3 4 5 7 2 3        | 3 4 5 7 2 3 Jolly Jumper            |
| 2 4 6 1 7 12 24 21 | 2 4 6 1 7 12 24X 21 No Jolly Jumper |
| 3 10 23 43 2 12    | 3 10X 23X 43X 2X 12 No Jolly Jumper |

18. Se desea tener una aplicación que permita registrar las solicitudes de una tienda que solo atiende a clientes vía correo electrónico.

Los pedidos que llegan deben tener un cliente, dirección - zona y una lista de ítemes que se pide y cada ítem debería tener un peso que defina su posibilidad de atención y en base a la suma de los pesos de los objetos de los pedidos se define el orden de atención de estos pedidos. Los pedidos jamás pesan 100 pto o más. y lo mínimo que pesan es 10 pto.

Las solicitudes se organizan de acuerdo al peso que tienen, del 10 al 19, del 20 al 29, y así sucesivamente del 90 al 99. Cada vez que se atiende el pedido, se debe a su vez registrar el pedido de acuerdo a la dirección del cliente.

Al finalizar el día, la tienda saca un listado que tiene los pedidos organizados por dirección - zona, cosa que al día siguiente su personal de reparto se hará cargo de la entrega de un conjunto de pedidos, se sabe que el personal de reparto se divide el trabajo por zonas.

- Elige la(s) estructura(s) de datos más adecuada(s) para resolver el problema.
  - Describe la estructura de los elementos (T) que se almacenan en la estructura de datos elegida.
  - Escribe el(los) método(s) necesario(s) para poder atender el requerimiento de un cliente y registrarlo para su posterior entrega. Debes cuidar que la estructura que elijas debe ser lo más eficiente posible.
  - Escribe el(los) método(s) necesario(s) para poder encontrar la lista que permita planificar las entregas cada día.
19. Se ha desarrollado una estructura de datos lineal que pretende mejorar la búsqueda de un dato. Esta estructura se llama ListaHilada, que es básicamente de un solo enlace al siguiente elemento, pero cada cuatro elementos éstos están enlazados de manera también simple hacia adelante. El modelo de la ListaHilada se muestra en la Figura 3.1.
- Un ejemplo de cambio de estado en la ListaHilada haciendo inserciones se muestra en la Figura 3.2.
- Todos los elementos en la ListaHilada se tienen ordenados ascendentemente.
- En el **Código 1** se tiene detalle de los atributos y operaciones de la ListaHilada.



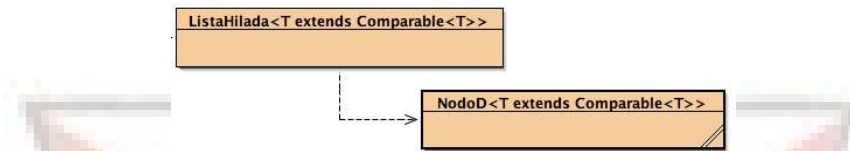


Figura 3.1: Modelo ListaEnlazada

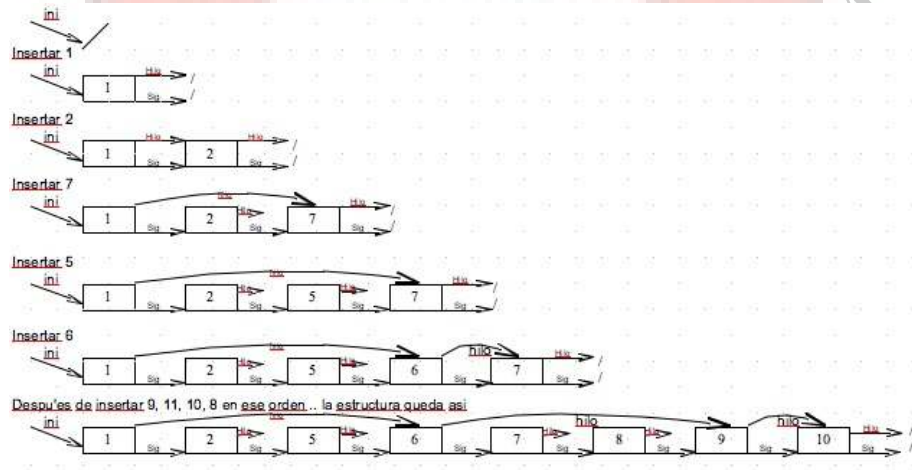


Figura 3.2: Ejecución de inserciones en ListaHilada

```

public class ListaHilada<T extends Comparable<T>>
{
 private NodoD<T> ini;
 public ListaHilada()
 { ini = null; }
 public boolean vacia()
 { return ini == null; }
 public void insertar(T d)
 {
 NodoD<T> p = new NodoD<T>(d);
 if (vacía()) ini = p;
 else
 insertar(ini, p, 1, ini);}
 private void insertar(NodoD<T> q, NodoD<T> p, int i, NodoD<T> h)
 {
 // metodo que inserta a p caminando con q
 // y dependiendo del estado del hilo de h
 // y la posicion de i se actualiza el hilo}
 public T buscar(T d)
 {
 // HACER ESTO}
 }
}

public class NodoD<T extends Comparable<T>> implements Comparable<NodoD<T>>
{
 private T dato;
 private NodoD<T> sig, hilo;
 public NodoD(T d){...}
}

```

```

public NodoD<T> sig(){...}
public NodoD<T> hilo(){...}
public void sig(NodoD<T> s){...}
public void hilo(NodoD<T> s){...}
public void dato(T s){...}
public T dato(){...}
public int compareTo(NodoD<T> otro){...}
public String toString(){...}
}

```

Realiza el método **T buscar(T d)**, que busca un elemento y lo devuelve si lo encuentra, caso contrario devuelve *null*. El método debe ser eficiente, aprovecha los enlaces de los hilos....

20. *Lic. R. Salazar* La sombra de un ArbolBB de tamaño **d**, es aquel ArbolBB que puede formarse considerando los elementos que se encuentran hasta el nivel **d**. Realiza un método **ArbolBB< T > sombra(ArbolBB< T > orig, int d)**, que construya la sombra del ArbolBB **orig** dado que el tamaño de la sombra es **d**.
21. Una librería quiere tener almacenados los ítemes que tiene a la venta. Cada uno de estos ítemes está dentro de una clasificación y tiene precio por menor y por mayor, además de la unidad. Por ejemplo: hojas existe una variedad, pero está clasificada como hojas. Las hojas por paquete tiene un precio y otro por caja. En este ejemplo el ítem es hojas, la descripción podría ser "de color" la unidad de venta por menor es paquete y la unidad de venta al por mayor es caja.  
La librería quiere tener velocidad cuando pregunta acerca de un ítem y su precio, por supuesto, el programa debería indicar las posibilidades de oferta que tiene para vender.
  - a) Elige la estructura de datos más adecuada para representar el problema.
  - b) Describe la estructura de los elementos que viven en la estructura de datos elegida.
  - c) Escribe el (los) método(s) necesario(s) para poder atender el requerimiento de un cliente. Por ejemplo: hojas bond oficio.
22. Dada una cadena, se pide encontrar el patrón de composición reducido correspondiente. Por ejemplo: si se tiene *aaaabbbbccdebbbfss* el patrón reducido será *a4b4c2d1e1b3f2s2*.
  - a) Elige la(s) estructura(s) de datos **MÁS ADECUADAS** para representar el problema.
  - b) Describe la estructura de los elementos que viven en la estructura de datos elegida.
  - c) Escribe el (los) método(s) necesario(s) para dada una cadena encontrar la cadena patrón reducida.
23. Dada una Lista Circular de Simple Enlace tipificada como: *ListaCSE < Integer >*, se pide separar los elementos en dos Listas Circulares de Simple Enlace, dado un dato **n**, de acuerdo al siguiente criterio, todos los elementos menores a **n** se ponen en una de las listas y los que no lo son, en la otra.
24. Dada una torre que contiene elementos de dos tipos, se necesita organizar los elementos de la misma, de forma tal que siempre existan elementos intercalados. Se tiene como condición de manejo que jamás se ingresa tres elementos del mismo tipo consecutivamente. La torre esta implementada en base a una pila, realice el (los) método(s) necesario(s) que permitan ingresar elementos a la torre de acuerdo a las especificaciones descritas.
25. La asignación de memoria es uno de los problemas más interesantes que existen dentro el desarrollo de los sistemas operativos y el mecanismo de GARBAGE COLLECTION también lo es. El GARBAGE COLLECTION consiste en liberar espacio de memoria que no es utilizado en un espacio de tiempo X.



En general la forma de organizar la memoria es por “bloques” de UN tamaño específico. Cada que se quiere “utilizar memoria”, se busca si hay un bloque que permita atender la solicitud.

La solicitud de asignación indica que ha fallado solamente si ningún bloque tiene espacio, caso contrario se dice que ha sido exitosa. Cuando una asignación ha sido exitosa, solamente permanece en la memoria si se lo usa en un periodo no mayor a X (por ejemplo 10). Por supuesto hay que cuidar que el espacio libre del bloque sea consistente después de una asignación. Se tienen tres datos muy importantes: a) el tamaño de una solicitud siempre es menor que el tamaño de un bloque, b) que en un bloque jamás se asignan más de dos solicitudes y c) nunca dos solicitudes usando la memoria tienen el mismo tamaño.

Del mismo modo cuando se requiere acceder a un bloque, se indica que ha tenido éxito solamente si el bloque está asignado con la solicitud buscada y si el tiempo de almacenamiento no ha pasado, caso contrario se dice que el requerimiento ha fallado.

Un dato importante es que jamás el bloque queda asignado por más de X periodos de tiempo sin haberlo utilizado, de tal manera que a tiempo de asignar el bloque se necesita indicar el tiempo en el que se realizó la solicitud, del mismo modo cuando se solicita acceder a un bloque se debe indicar el tiempo en el que se está intentando hacer el acceso.

Por ejemplo, si se supone que se tiene una memoria organizada en 15 bloques cada uno de 500 de tamaño, todas libres al inicio y si se tiene un requerimiento de asignación que tiene el siguiente formato:

1 200

donde 1 es el tiempo en el que se pide asignar algún bloque de memoria y 200 es el tamaño de la solicitud. La solicitud será exitosa e informa en que bloque se asignó, en este caso se asigna en el primer bloque.

Si se tiene un requerimiento de acceso que tiene el siguiente formato:

7 10 200

donde 7 significa el tiempo en el que se requiere el acceso, 10 es el número de bloque que se quiere acceder y 200 es el tamaño de la solicitud, debería indicar que el requerimiento falló ya que en ese bloque no hay nada asignado.

Si el requerimiento de acceso fuera:

7 1 200

el acceso es exitoso; ya que no ha pasado aún X (10) periodos de tiempo y el bloque 1 está asignado con la solicitud 200.

Si el requerimiento de acceso fuera:

12 1 200

el acceso es exitoso; ya que no ha pasado más de X (10) periodos de tiempo desde su asignación y/o último acceso

Pero si el requerimiento de acceso fuera:

27 1 200

el acceso falló; ya que ha pasado más de X (10) periodos de tiempo desde su asignación y/o último acceso y el bloque 1 ya habría sido liberado automáticamente (GARBAGE COLLECTION).

Entonces si se tiene la siguiente secuencia de uso de memoria, considerando por ejemplo 15 bloques de 500:

| ENTRADA  | RESULTADO |
|----------|-----------|
| 1 200    | exito     |
| 2 300    | exito     |
| 3 400    | exito     |
| 5 250    | exito     |
| 6 350    | exito     |
| 9 1 200  | exito     |
| 13 1 250 | fallo     |
| 20 2 300 | fallo     |
| 21 1 200 | fallo     |
| 22 1 250 | fallo     |
| 22 150   | exito     |
| 23 1 200 | fallo     |
| 24 1 300 | fallo     |
| 25 1 150 | exito     |
| 30 2 300 | fallo     |
| 31 3 400 | fallo     |

Por supuesto, como se puede observar en el ejemplo, las solicitudes se hacen en orden cronológico. Se puede asumir que los números de bloque siempre son correctos.

- Elige la estructura de datos que más se adecue para resolver el problema.
- Describe la estructura de los elementos que residirán en la estructura de datos que elegiste (realiza el modelo).
- Escribe el (los) método(s) que permitan asignar memoria a una solicitud.
- Escribe el (los) método(s) que permitan acceder a un bloque de memoria.

26. En Las Vegas un juego muy popular es el blackjack que consiste en jugar con a lo sumo tres cartas contra el croupier (persona que reparte cartas y es personal del casino); el objetivo es que las cartas sumadas se aproximen sin sobrepasar a 21. Para ello la banca (se llama así a la mesa en la que se juega considerando un maso de cartas completas) reparte 2 cartas a cada jugador incluyendo al crupier; los valores de las cartas son: el as vale 1 u 11, las figuras valen 10, y las cartas numéricas su valor natural; pudiendo cambiar el valor del as según la jugada.

El jugador después de repartir las dos cartas tiene la posibilidad de plantarse (quedarse con las cartas que tiene) o pedir carta, sin pasarse del 21 ya que pierde automáticamente.

Gana finalmente el que tenga el número más alto, cercano al 21, o saque un blackjack (sea igual a 21). Por su parte el croupier tiene reglas rígidas a las que atenerse: si su puntuación inicial fuera 16 o menor está obligado a tomar otra carta, y se plantará siempre que su puntuación alcance 17 o mayor. Estas reglas las aplicará independientemente de las jugadas que tengan cada uno de los jugadores.

El maso de cartas original, está formado por 'n' cartas las mismas que están barajadas de forma aleatoria; de este maso se va repartiendo cartas a los 'm' jugadores del juego. Jamás las cartas van saliendo del medio del maso siempre salen de la parte de arriba del maso; que tiene las cartas volcadas para mantener la emoción del juego.

Las cartas que hay en el maso son: (1 (as), 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K) cada uno de ellos en cuatro palos diferentes (C(corazon), D(diamante), T(trebol), E(espada)), en total existen 52 cartas en el maso original.

Entonces en este juego hay dos rondas: la primera que permite repartir dos cartas a los jugadores, la segunda que permite repartir una carta más a los que deseen, salvo el croupier.

Después de estas dos rondas se decide quién es el ganador del juego.

- Elige las estructuras de datos adecuadas para resolver el problema.
- Describe la estructura de los datos que contendrán tus estructuras de datos.
- Escribe el(los) método(s) necesario(s) para realizar la repartija en la primera ronda a los 'm' jugadores, no olvides al croupier.
- Escribe el(los) método(s) necesario(s) para realizar la repartija en la segunda ronda a los 'm' jugadores, no olvides que las reglas del croupier son específicas y fijas en esta ronda. Para el resto de los jugadores puedes asumir que la solicitud de una carta más es aleatoria.

27. Se tienen las estructuras de datos **p**, **b**, **a**, **l** que son una pila de enteros, una bicola de enteros, un árbol binario de búsqueda de enteros y una lista de enteros, respectivamente:

```
Pila<Integer> p = new Pila<Integer>();
BiCola<Integer> b = new BiCola<Integer>();
ArbolBB<Integer> a = new ArbolBB<Integer>();
Lista<Integer> l = new ListaSE<Integer>();
```

Considera las siguientes instrucciones:

```
1 a.insertar(16);
2 a.insertar(20);
3 a.insertar(10);
4 a.insertar(40);
5 a.insertar(30);
6 vaciar(a.preorden(), p);
7 vaciar(p, b);
8 vaciar(b, a);
9 l = a.inorden();
```

El método vaciar es uno que permite pasar los elementos de una estructura (origen) a otra (destino), destruyendo la origen. Puedes asumir que su comportamiento es correcto.

- Muestra “graficamente” (dibuja), el estado de las estructuras **a**, **p**, **b**, **l** a medida que se realizan los procesos.
  - ¿La pila **p** está ordenada después de la instrucción 7?
  - Al terminar el proceso (en la instrucción 9), ¿el árbol **a** es el mismo en estructura que antes de los procesos vaciar?
28. En un juego denominado “*¿hola quien llama? - la llama que llama*”, se ha planeado registrar todos los mensajes en un sistema. Lo que se desea es saber *qué llama llama a que llama*. Todos los registros tienen un orden de aparición y se quiere saber *¿cuál es la llama que llama?*, por lo que la posición de registro es importante. Por ejemplo, si se tienen los siguientes registros:

```
hola?
hola?
hola?
quien es la llama que llama?
quien es la llama que llama?
hola?
quien es la llama que llama?
quien es la llama que llama?
```

Se tendrá el siguiente reporte:

```
La llama 3
La llama 2
La llama 4
La llama 1
```

Considerando este ambito de aplicación:

- Elige la estructura de datos que se adecua mas para resolver el problema de manera eficiente
- Explica/define cuál es la estructura de los elementos que contendrá tu estructura de datos
- Escribe el (los) método(s) necesario(s) para permitir dado una lista de registros, definir *¿qué llama llama?*

29. Se quiere aumentar una operación a toda lista, que consiste en hacer un *split – division* de la lista en otras mas pequeñas de tamaño  $n$ . Por ejemplo si se tiene la lista:

```
{2, 4, 5, 7, 2, 8, 10, 14, 19, 23, 59, 10, 14}
```

Y se quiere hacer un split de 2, se tendría el siguiente resultado:

```
{{2, 4}, {5, 7}, {2, 8}, {10, 14}, {19, 23}, {59, 10}, {14}}
```

Para realizar esta tarea puedes elegir la lista de tu preferencia: **ListaSE**, **ListaDE**, **ListaCSE** o **ListaCDE**. Lo único que debes cuidar es que si por ejemplo la lista original a la que le pides *split(n)*, es una ListaDE, entonces la lista resultante también lo será, además de cada una de las listas que aparecen en el resultado.

30. El manejo de los objetos en jdk, en particular, considera conceptos de asignación de memoria dinámica que puede ocasionar algunos funcionamientos “raros”. Cuando una Cola de Prioridades permite realizar la operación *ver()* retorna el dato del frente de la cola y desde donde lo solicitaron es posible cambiar el dato, por ejemplo observa el siguiente código:

```
1 ColaPrioridad<Estudiante> cola = new ColaPrioridad<Estudiante>();
2 Estudiante est;
3 est = new Estudiante("Luis", 50);
4 cola.encolar(est);
5 est = new Estudiante("Maria", 30);
6 cola.encolar(est);
7 est = new Estudiante("Sara", 100);
8 cola.encolar(est);
```

```
9 est = new Estudiante("Jose", 10);
10 cola.encolar(est);
11 est = cola.ver();
12 est.setNota(50);
```

Para ubicar un estudiante en la cola, se utiliza su nota como prioridad. En la instrucción 11, se recupera el estudiante del frente y en la 12 se ha cambiado su información que es además su prioridad!!. ¿cómo queda la cola? ¿Sigue respetando la prioridad? Arregla lo que consideres necesario para que dinámicamente la cola pueda mantenerse siempre consistente, a pesar de que un dato que aun esta en la cola cambie.

31. Dada una Pila de elementos ordenables se te pide obtener los elementos en otra pila, de tal manera que los elementos queden ordenados de manera descendente, es decir el elemento mas pequeño en el tope de la pila. Recuerda no debes alterar las prestaciones de servicio de las pila. Es posible apoyar con estructuras que consideres necesarias tu solución. Hay que siempre considerar la mejor opción.

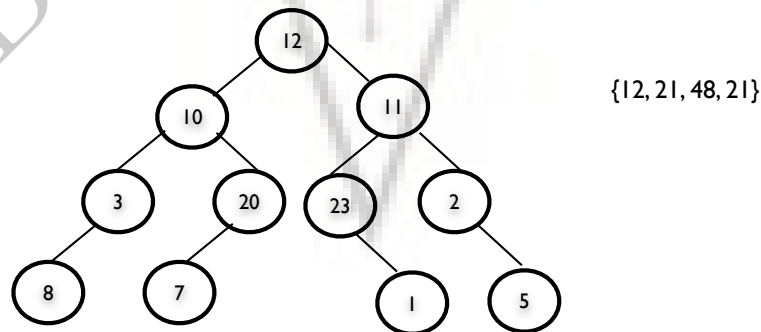
## Capítulo 4

# Estructuras de Datos No Lineales

Para todos los ejercicios debes utilizar adecuadamente Estructuras de Datos.

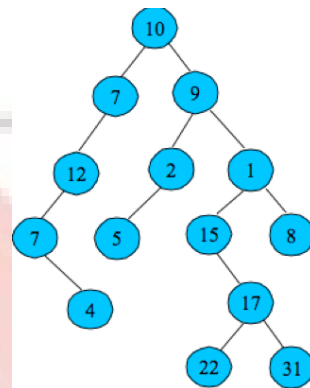
1. Dado un árbol Binario indique si este es balanceado. Se dice que un árbol es balanceado si la altura de la descendencia izquierda difiere en a lo sumo 1 de la altura de la descendencia derecha.
2. Dado un árbol binario se desea mostrar los elementos de la estructura de la siguiente manera: siempre empezar por el último nivel e ir mostrando primero el elemento más a la izquierda del nivel, luego el que esta más a la derecha del nivel; luego el segundo más a la izquierda y después el segundo más a la derecha, y así sucesivamente hasta agotar todos lo elementos del nivel y cubrir todos los niveles del árbol.
3. Dados dos árboles binarios se desea saber si son semejantes. Se dice que dos árboles son semejantes cuando estructuralmente ambos son iguales sin importar su contenido.
4. Se desea saber cuántos nodos de un árbol binario tienen solamente una descendencia.
5. Suponga que el árbol binario tiene datos que son comparables, emita un reporte de los elementos menores por cada nivel del árbol.
6. Suponga que el árbol binario tiene datos que son comparables, emita un reporte de los elementos menores por cada rama del árbol.
7. Se desea saber si en un árbol binario todos sus nodos son completos.
8. Dado un árbol binario de búsqueda, se desea encontrar el árbol binario de modo que el árbol resultante sea el resultado de hacer un **repmin** en el árbol binario de búsqueda. Se dice que el proceso **repmin**, consiste en encontrar el elemento mínimo del árbol y reemplazarlo en los nodos hoja.
9. Realice el constructor que permita en base a una cadena que contiene una expresión en forma prefija, construir un árbol binario de expresión correcto.
10. Dado un árbol binario de búsqueda, se quiere recoger los elementos que estén en el intervalo  $[x, y]$ , de forma eficiente.
11. Dado un árbol binario de búsqueda, se quiere recoger los elementos del camino de **x** a **y**.
12. Dado un árbol binario, se quiere recoger los elementos del camino de **x** a **y**.
13. Realice el método **evaluar** de un árbol binario de expresión.

14. Diseña un árbol que permita representar una expresión no necesariamente binaria, es decir, representar el operador **no lógico**, el **raíz cuadrada**, etc. o representar el operador ternario **if\_then\_else** que tiene el formato *cond?cuerpoSI : cuerpoNO*
15. Un país no posee de carreteras la única vía de transporte es la fluvial. Este país se comunica a través de los ríos, y los pueblos están a la rivera de los ríos. Todos los ríos desembocan en el mar. Se pide determinar cuáles pueblos están además de la rivera del río a orillas del mar. Dado, que desde la capital del país, solamente existe una ruta hacia cada pueblo del mismo, se desea además encontrar el camino para llegar de un pueblo a otro por las orillas del mar.
16. Dado un conjunto de pares ordenados, indique si es posible que esta información forme un árbol. Estos pares ordenados, tienen la intención de ser pares ordenados de la forma: (**padre, hijo**), entonces el par ordenado (**5,6**) significa que el padre es 5 y el hijo es 6. Hay que recordar que un elemento no tiene padre, y es el elemento raíz del árbol. Por ejemplo, los pares  $\{(3,5), (3,7), (3,8), (7,1), (7,2), (8,10)\}$  forman un árbol, cuya raíz es el 3, los elementos del nivel 1 son: 5, 7 y 8; y los del nivel 2 son: 1, 2 y 10. En cambio, los pares  $\{(10,2), (2,5), (5,10), (2,6), (6,7), (7,9), (10,3), (10,9)\}$  no forman la información de un árbol.  
Con la información de entrada como una lista de pares ordenados, construye el árbol que corresponde si fuera posible, caso contrario, se debería crear un árbol vacío.
17. Se quiere modelar una red con una topología en forma estrella. Todos los puntos se conectan a un punto central de distribución y desde allí hacia los distintos puntos de la red. Dada una ruta se desea saber si es una ruta posible dentro la red.
  - Elige la estructura de datos adecuada para resolver y representar el problema.
  - Describe la estructura de los elementos que residirán en la estructura de datos que elegiste.
  - Escribe el (los) método(s) necesario(s) para saber si una ruta dada por entrada es válida en la red. Además debes indicar en que punto, si hubiera, existe una ruptura de la red.
18. Considera un árbol binario en el que se almacenan datos que pueden ser reducibles, se pide encontrar la lista de reducibles del árbol por niveles. Para ello realiza el proceso *ListaSE < Reducible > reducidos(ArbolBin < Reducible > a, Reductor r)*, como notarás en este caso se requiere de información adicional que es un *Reductor r*, cuya misión es dado un Reducible encontrar otro o dados dos Reducibles encontrar otro. Por ejemplo si se tiene el siguiente árbol de enteros y el factor de reducción es la suma, se tendrá el siguiente resultado.



19. Dado un árbol binario, en el que se tienen elementos que son de tipo **entero**, se pide encontrar todos los elementos múltiplos de un dato **n** dado como parámetro de entrada. La Figura 4.11 muestra un ejemplo del resultado del proceso.





Si se tiene  $n = 3$   
 $\{9, 12, 15\}$

Figura 4.1: Ejemplo

20. Dado un Arbol Binario, se quiere dar la vuelta la rama más pequeña del árbol. Por ejemplo observe la figura 4.2, muestra un ejemplo particular del resultado de este trabajo.



Figura 4.2: Ejemplo

21. Dado un Arbol Binario de Búsqueda de números enteros, se quiere encontrar la mediana de cada nivel. Por ejemplo observe la figura 4.3, muestra un ejemplo particular del resultado de este trabajo. Debes considerar la solución más eficiente para realizar esta tarea.
22. Se tiene en un Arbol Binario representado un mapa de rutas fluviales de una cuenca. Esta cuenca empieza en un punto, a partir de este punto se van ubicando pueblos a las orillas de los rios que a su vez son puntos de bifurcación en dos rios.... la información que se tiene de cada punto en el mapa es el nombre del pueblo y el tiempo que toma llegar a él desde el anterior pueblo. Un viajero quiere conocer los pueblos más lejanos a los que pueda llegar, el gran problema de este viajero es el tiempo, él solamente cuenta con  $X$  cantidad de tiempo y quisiera en ese tiempo conocer la mayor cantidad de pueblos que pueda.

Por ejemplo de la Figura 4.4, se muestra un mapa de la **cuenca "Bora"**: Si un viajero con tiempo  $X = 120$  quisiera la ayuda para decidir cual es la mejor opción de recorrido. Para este requerimiento la respuesta vendría a ser:

{Notar, NejoCo, DillaR, Onom}



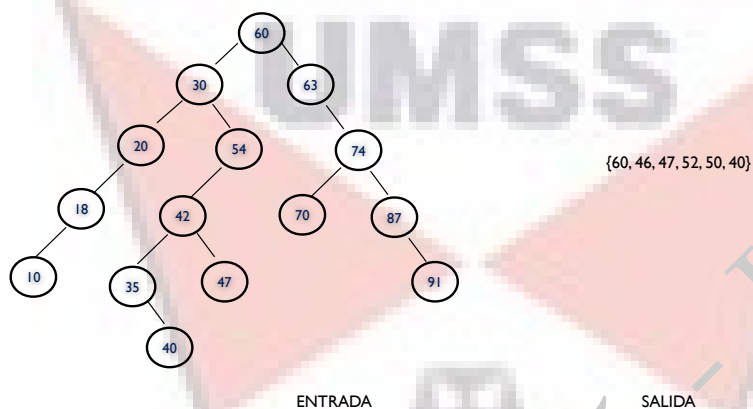


Figura 4.3: Ejemplo

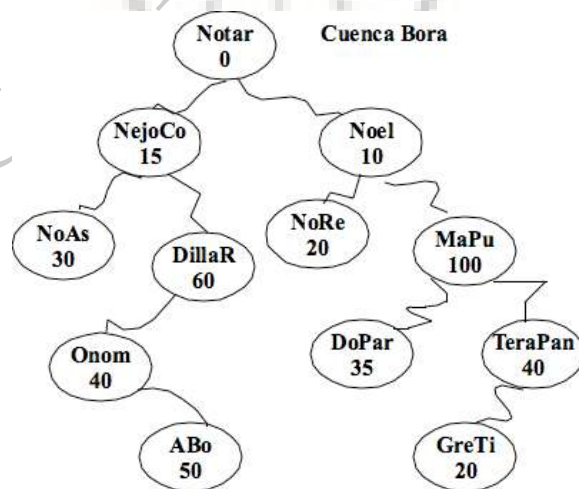


Figura 4.4: Mapa de Cuenca Bora

Las otras opciones no permiten al viajero conocer la mayor cantidad de pueblos.

- a) Realiza el modelo adecuado que describa el problema.
  - b) Realiza el (los) método(s) necesario(s) para ayudar a un viajero a encontrar cual es la ruta que le permita visitar la mayor cantidad de pueblos dado que tiene un tiempo **X**.
23. Se tiene el país de los **"PSK2"** lleno de tesoros, los que son una atracción para los piratas de la zona. Los ciudadanos del país se han preparado bien a las invasiones de los piratas, es así que han puesto fuertes en sus regiones mediterráneas, ya que ellos saben que en las costas del mar, les es imposible hacer frente a los piratas, ya que ellos invaden a **"PSK2"** por la rivera del mar. Los piratas han decidido no correr demasiados riesgos, por lo que su invasión la planifican desde el puerto mas seguro y que les permita llegar en un viaje de ida y vuelta a la mayor cantidad de regiones con menor riesgo. De cada región mediterránea se conoce el número de personas en sus fuertes y el monto del tesoro de la región, se dice que es una región de poco riesgo para los piratas si el numero de personas en el fuerte no los sobrepasa. Tu tarea es ayudar a los piratas a decidir la ruta mas conveniente de manera que no corran riesgos y que además indique la cantidad de tesoro acumulado en la ruta. Es posible que no exista tal ruta, por lo que en ese caso debes emitir un reporte **"IMPOSIBLE"**. Se sabe que **"PSK2"** es un país bien conectado, es decir siempre hay **SÓLO UNA** ruta entre una región y otra del país, también se sabe que todas la regiones siempre pueden desembocar en el mar. Un dato importante, es que los piratas deben empezar y terminar su invasión en el mismo puerto, por obvias razones de transporte.
- a) elige la estructura de datos mas adecuada para resolver el problema
  - b) elige la estructura de los elementos que se almacenarán en tu estructura de datos
  - c) escribe el (los) método(s) necesario(s) para indicar a los piratas la ruta más conveniente de invasión a **"PSK2"**
24. Dado una Arbol Binario de Búsqueda, se pide mostrar los elementos del árbol intercalados, es decir: el primer menor, el primer mayor, el segundo menor, el segundo mayor, y así sucesivamente.....
25. Se quiere modelar el mundial de futbol a partir de los octavos de final. Para ellos se deben registrar los países que se enfrentan, quiénes ganan, quiénes pasan a cuartos de final y quién es el campeón del mundo. Además de registrar esta información se requiere tener los datos de los goles que cada selección en cada partido ha anotado y también los goles en contra.
- Al terminar el campeonato, se desea saber la lista de los equipos que llegaron a los octavos de final. La selección que ha anotado más goles, la selección que ha recibido mas goles en contra.
- Debes considerar que a partir de octavos de final los partidos definen a sus ganadores para pasar a la siguiente fase. Es muy importante también que las selecciones recuerden los datos de los goles a favor y en contra al llegar a los octavos de final, y a partir de ese momento es importante que se registre los goles a favor y en contra por partido jugado.
- Dada esta información:
- a) elija la estructura de datos que más se adecua para resolver el problema
  - b) describe la estructura de los elementos de la estructura de datos que elegiste
  - c) escribe el (los) metodo(s) necesarios para saber qué equipos estan en octavos de final
  - d) escribe el(los) métodos necesarios para saber que selección ha anotado más goles, indicando el detalle de los partidos en los cuales los ha obtenido
26. Se tiene un árbol binario, en el que se tienen elementos que son de tipo **entero**, encuentre la lista de la suma de los elementos por nivel, que tiene el árbol. Considere el siguiente ejemplo cuyo resultado del proceso se observa en la parte derecha de la Figura 4.11.

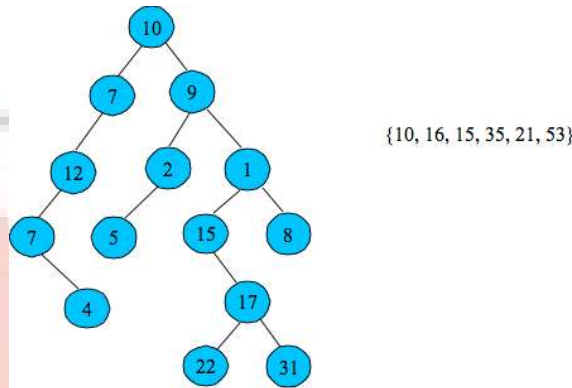


Figura 4.5: Ejemplo

27. Dado un árbol binario de caracteres ordenado, listar todas las hojas del mismo, este proceso se repite hasta terminar vaciando el árbol.

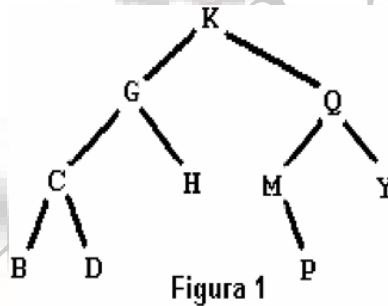


Figura 4.6: Ejemplo

Si se toma el árbol de la figura 4.6, el resultados sería

BDHPY  
CM  
GQ  
K

28. Dado un Árbol Binario, realice el (los) método(s) necesario(s), para poder hacer un **"mirror"** del árbol. Es decir, todo lo que se encontraba en descendencia izquierda ahora será descendencia derecha. Por ejemplo, en la figura 4.7 se observa la ejecución de **"mirror"** sobre el árbol **A**.
29. Implementar una aplicación que permita:
- Crear un árbol binario de búsqueda con datos ingresados en una lista. La lista contiene números enteros.
  - Almacenar en una estructura adecuada la rama más larga del árbol.
  - Encontrar la lista de elementos que son mayores a la media aritmética de los elementos del árbol. La media aritmética se define como:  $m = (min + max)/2$

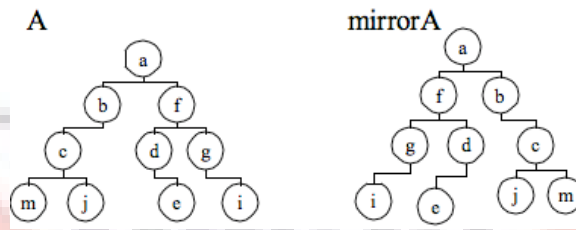


Figura 4.7: Ejemplo de aplicación de mirror

30. En una aldea, por decisión de las autoridades han decidido sembrar en cada región un tipo de tubérculo. Es así, que por ejemplo la región **A** sólo siembra **Oca** y así lo hacen todas las regiones. Desde la capital de esta aldea, se tiene una sola forma de llegar a cada región. Se ha tenido un embate climático muy fuerte, y el responsable de economía ha dispuesto y encargado que se le dé un reporte del camino más productivo de la aldea. Se dice que el camino más productivo es aquel en el que puede cosechar la mayor diversidad de tubérculos.

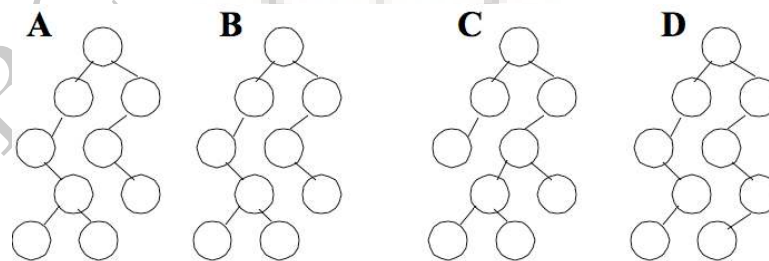
- Elige la estructura **MAS ADECUADA** que permita modelar y representar el problema
- Describe la **ESTRUCTURA** de los elementos que residirán en la estructura de datos que elegiste en el anterior inciso
- Define el (los) métodos necesarios para responder el requerimiento del responsable de economía de la aldea

31. Dados dos árboles binarios de elementos de tipo **T** que son **fusionables**, se desea encontrar el árbol binario fusionado si es posible.

Se dice que un elemento es **fusionable** ya que pueden unirse a otros elementos de tipo **T**, a través de un método cuya firma es: **T fusionar(T)**

Se dice que se puede fusionar dos árboles de elementos fusionables, si son semejantes.

Se dice que dos árboles son semejantes, si tienen la misma estructura (ver Figura 4.8)



A y B Semejantes

C y D No Semejantes

Figura 4.8: Ejemplo de semejanza de árboles

En caso de que no sean semejantes, la respuesta debería ser un árbol binario vacío.

32. Dado un Arbol Binario de caracteres, se pide encontrar una lista que contenga, las hojas que tiene el árbol por niveles. Por ejemplo dado el arbol de la figura 4.9

Del Arbol A el resultado seria:

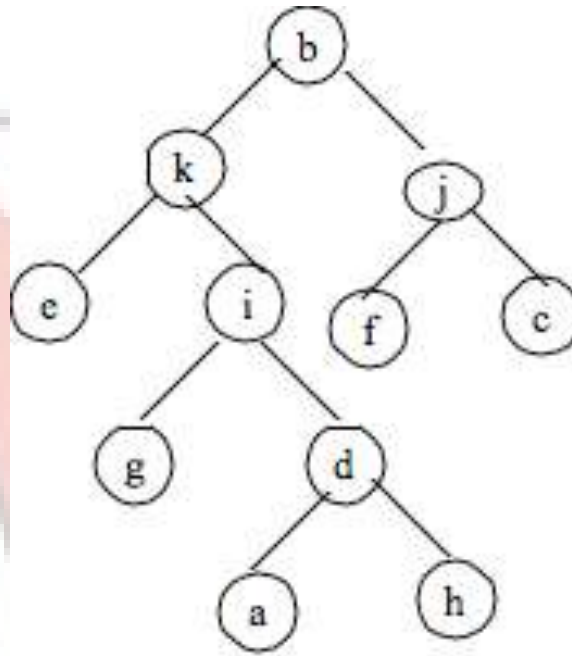


Figura 4.9: Arbol A

$\{\{\}, \{\}, \{e, f, c\}, \{g\}, \{a, h\}\}$

33. Dados **A** y **B** árboles binarios, se requiere saber si cumplen la condición de **igualdad de contenido**. Se dice que dos árboles tienen igualdad de contenido si sus datos en contenido son los mismos y además éstos están en el mismo nivel en ambos árboles.

*Condiciones:* dos árboles binarios son iguales si tienen el mismo nivel y los elementos por nivel son iguales, no importa el orden. Entonces, se averigua si tienen mismo nivel si es así se empieza a probar los elementos nivel por nivel, basta que encuentre un nivel en el que los elementos no sean iguales entonces los árboles no son iguales.

Se da por hecho que los árboles saben dar su nivel, sin embargo no saben dar sus elementos por niveles. También se da por hecho que las listas saben responder de forma correcta a la pregunta de si es igual de contenido con otra.

- Define el método en la clase Arbol Binario, que permita recoger los elementos de un nivel **N**
  - Escribe el método **iguales** en una clase aparte que reciba dos árboles binarios **A** y **B** y decida si cumplen la condición de igualdad de contenido.
34. Se tiene un Arbol Binario A, se desea saber si A tiene una podación copa, se dice que un árbol binario tiene podación copa si para todas las ramas del árbol se cumple que tiene por lo menos la mitad de los nodos de la rama más larga. Realice el(los) método(s) que permitan decidir si un Arbol Binario tiene podación copa.
35. Se tiene un Árbol Binario de Búsqueda A de números enteros, se desea saber si el árbol tienen por lo menos una secuencia creciente seguida en pasos d. Se dice que B es una secuencia creciente seguida en pasos d si se da lo siguiente:  $B = b_1, b_2, b_3, b_4, \dots, b_n$  y para todo  $b_i, b_{i+1}$  que pertenece a B, se cumple que

$b_{i+1} - b_i = d$ . Escriba el (los) método(s) necesario(s) para saber si A tiene por lo menos alguna secuencia creciente seguida en pasos d.

36. Se tiene un Arbol Binario A, se desea saber si A tiene una podación simétrica, se dice que un árbol binario tiene podación simétrica si para todos los niveles del arbol se cumple que tiene por lo menos la mitad de los nodos permisibles. Se sabe que la cantidad máxima de nodos permisibles en un nivel es  $2^{niv}$ . Realice el(los) metodo(s) que permitan decidir si un Arbol Binario tiene podación simétrica.
37. Se tiene un Arbol Binario A, se desea saber si este árbol es también balanceado por compensación. Un arbol binario es balanceado por compensación si la altura alterna de sus subarboles difiere en a lo sumo 1, es decir si en el árbol el subarbol derecho es mas alto (a lo sumo con uno) que el subarbol izquierdo, entonces en estos subarboles correspondientes deberían los respectivos subarboles izquierdos ser mas altos que los derechos, y en la próxima validación nuevamente los derechos y luego los izquierdos y así sucesivamente... Realiza el (los ) método(s) necesarios para poder decidir si un árbol binario es balanceado por compensación
38. Dado dos árboles binarios de enteros A y B, se desea saber si ellos son **sempeso n**. Se dice que dos árboles son **sempeso n**, si tienen la misma estructura y además la suma de sus elementos difiere en a lo sumo **n**. Por ejemplo, considerando la Figura 4.10; los árboles A1 y B1 son **sempeso** para  $n = 10$ , pero los árboles A1 y C1 no lo son (son semejantes en estructura pero no cumplen la condición del peso) y tampoco A1 y D1 (cumplen la condición del peso pero no de la semejanza de estructura).

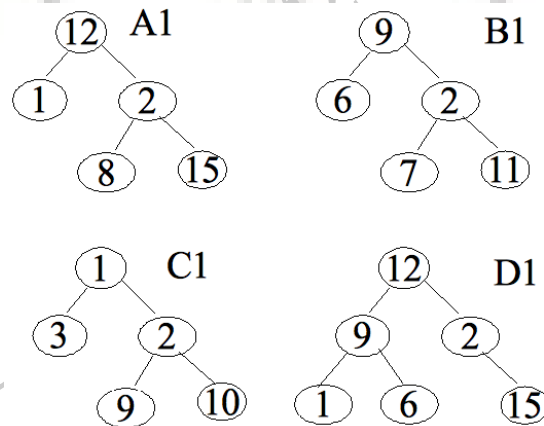


Figura 4.10: Ejemplo de arboles con sempeso

39. Un principe está en edad de contraer nupcias, para ello necesita encontrar a una princesa; por lo que, pide a su asesor realizar la tarea de encontrar a la afortunada, pero... el principe tiene condiciones que debe considerar el asesor: a) tiene una comarca definida en la cual se buscará a la princesa, b) no está dispuesto a arriesgar su vida y c) no tiene la mínima intención de recorrer caminos largos. El asesor ha analizado la comarca indicada por el principe y ha notado que a todas las regiones de las mismas se puede llegar a partir de un punto inicial, y por fortuna tiene el dato de los lugares donde vive alguna princesa, así mismo los lugares donde habitan feroces dragones y regiones donde habitan bondadosas hadas. Con esta información el asesor debe cumplir la tarea que el principe le ha encomendado y retornar con los datos de la princesa que cumple con las condiciones del principe, en caso de que existiera.

a) Elige la estructura **MAS ADECUADA** que permita modelar y representar el problema

- b) Describe la **ESTRUCTURA** de los elementos que residirán en la estructura de datos que elegiste en el anterior inciso
- c) Define el (los) métodos necesarios para ayudar al asesor del palacio a responder el requerimiento del príncipe.
40. Se desea validar un discurso que tiene varias frases, para ello se ha dispuesto de un diccionario de palabras permitidas (sin contar acentos) y otro de palabras denominadas **stop**. Debido a que el discurso tiene mucha importancia se ha decidido quitar del discurso aquellas frases que contengan mas del 50 % de palabras no válidas.

Por ejemplo: si se tiene el diccionario de palabras válidas y el de **stop**:

| D I C C I O N A R I O |            | S T O P |     |     |     |
|-----------------------|------------|---------|-----|-----|-----|
| cerca                 | superacion | y       | es  | se  | les |
| ahora                 | constancia | o       | un  | su  |     |
| hora                  | corriente  | de      | una | sus |     |
| concurso              | tezon      | la      | esa | lo  |     |
| examen                | excluido   | el      | con | a   |     |
| competencia           | cantidad   | las     | que | en  |     |
| clasificacion         | valor      | los     | por | le  |     |

Y el siguiente discurso:

estimados competidores esta es una prueba que les permitira llegar mas alla de lo que esperan exito en sus emprendimientos adelante con tezon constancia esfuerzo y sacrificio

el tezon y esfuerzo que ustedes le ponen a la competencia es un examen a la hora de la clasificacion la constancia es camino a la superacion y ahora que se acerca el final de la competencia es hora de demostrar su valor constancia y superacion

La primera frase tiene 14 palabras no permitidas de las 16 que no son stop, por lo tanto debe ser desechada del discurso. La segunda frase tiene 6 palabras no permitidas de las 20 que no son stop, por lo tanto debe ser incluida en el discurso.

Sobre la base de esta descripción se pide:

- identificar las estructuras de datos MAS adecuadas para modelar el dominio del problema: diccionarios y discurso
  - definir de forma clara la estructura de los elementos que contendrán las estructuras de datos antes elegidas
  - implementar el (los) método(s) necesario(s) para dado un discurso, generar otro con las frases correctas, de acuerdo a descripción del problema
41. Los gigantes de acero, son robots luchadores que han generado un movimiento económico muy grande en las casas de apuesta y competencias de esta naturaleza. Sin embargo muchas de estas casas son ilegales y sólo las personas involucradas en el negocio saben de estos lugares. Existe una red bien montada de estas casas, las mismas que están distribuidas estratégicamente, de tal manera que para llegar de una casa a otra se tiene exactamente un camino. En estas casas, se tiene una lista de las personas que han apostado, han perdido y no han pagado. Por lo que cuando una persona desea apostar, lo primero que se hace es consultar los datos para verificar si no es una persona con deudas, si es así se deja a la persona apostar, pero si no es imposible que esta persona pueda realizar una apuesta en esa casa, es más.... es



posible que corra riesgo estando allí. Max, es un apostador empedernido que ha trazado una ruta válida de las casas en las que quiere jugar, pero también es una persona que no ha respetado sus deudas, por lo que debes ayudarlo a cuidarse de entrar en los sitios en los cuales debe dinero producto de anteriores apuestas. El programa debe decirle en que casas puede entrar a apostar. Para ellos Max sólo cuenta con los nombres de las casas.

Por ejemplo, si se tiene la siguiente información de las casas: de donde se llega, su nombre y la lista de deudores....

| ANTERIOR | NOMBRE  | LISTA DEUDORES                  |
|----------|---------|---------------------------------|
|          | Tron    | Max, Luis, Paco, Orian, America |
| Tron     | Masa    | Paco, Lucas, Orian, Viola       |
| Tron     | Mega    | Lucas, Violeta, Orian, Marcos   |
| Mega     | Kron    | Hugo, Orian, Laura              |
| Masa     | Tera    | Luis, Max, Orian, Cornelius     |
| Masa     | Klavera | Joana, Juan, Orian, Luis        |
| Klavera  | Dinamo  | Paco, Hugo, Orian, Luis         |
| Dinamo   | Klaus   | Lucas, Max, Orian               |
| Mega     | Zoo     | Luis, Orian, Joana              |
| Zoo      | Museo   | Carla, Orian, Carlos            |
| Kron     | Mauso   | Orian, Carla                    |

Y Max quiere realizar la siguiente ruta {Dinamo, Klavera, Masa, Tron, Mega, Zoo}, sólomente podrá entrar sin correr ningún riesgo a: {Klavera, Mega, Zoo}. Por supuesto, como es de esperarse es posible que la lista resultado sea vacia, ya que hay jugadores que ya no lo son, y más bien son embaucadores como Orian que ya no tiene pisada en ninguna de las casas.

En este caso, debes proveer de generalidad a tu programa de tal manera que dado el nombre de un apostador y una ruta tentativa, permita encontrar la lista de casas que puede visitar. Por razones estratégicas es imposible que las casas estén aglomeradas alrededor de una casa, por lo que partir de una casa sólo se puede llegar a dos casas y así sucesivamente.

Sobre la base de esta descripción se pide:

- identificar las estructuras de datos MAS adecuadas para modelar el dominio del problema.
- definir de forma clara la estructura de los elementos que contendrán las estructuras de datos antes elegidas
- implementar el (los) método(s) necesario(s) para dada una ruta tentativa y un apostador, encontrar la ruta sin riesgo.

**HINT:** las casas en las que NO puede entrar son aquellas en las que la cantidad de vecinos en los que tiene deuda son mayores o iguales a la cantidad de casas en las que no tiene deuda.

(ross)

- Dado un Árbol Binario de enteros, se te pide decidir si está ordenado en alguno de los ordenes de recorrido. Trata de hacer un proceso **EFICIENTE Y OPTIMO** para resolver el problema.
- Un equipo de desarrollo software, está trabajando en un nuevo sistema operativo y requiere implementar una técnica de almacenamiento que obliga al usuario a almacenar en una ruta desde la raíz **c:** de la computadora, considerando que jamás las bifurcaciones desde la raíz **c:** y los directorios de la PC pueden ser más de dos.



Cada vez que se guarda un archivo se almacena su respectivo identificador (nombre), su ruta de acceso y además el momento en el que fue almacenado; esto permite tener una cronología de almacenamiento.

La tarea que tiene este equipo de desarrollo de cabeza, es que deben plantear la inserción del archivo en la PC, considerando como criterio de ordenación el nombre del archivo. Por otro lado están presionados respecto al uso de memoria por lo cual la información de todos los archivos en este sistema operativo se deben almacenar en una sola estructura de datos.

Un otro requerimiento es que sobre la base de esta información se debe reportar los 'n' primeros archivos almacenados, los 'n' últimos archivos almacenados.

El equipo de desarrollo está lidiando con otros problemas más, propios del desarrollo de un sistema operativo, por lo que han solicitado que les ayudes en las tareas antes descritas; para ello:

- a) elige la estructura de datos que más se adecue para resolver el problema.
  - b) describe la estructura de los elementos que residirán en la estructura de datos que elegiste (realiza el modelo).
  - c) escribe el (los) método(s) que permitan encontrar los 'n' archivos más recientes. **NOTA** recuerda que el orden cronológico es único y que cada archivo conoce su orden cronológico de ingreso; por lo que el equipo espera que les brindes una solución ÓPTIMA mejor si se puede resolver en una sola pasada en la estructura. No confundas orden con tiempo
44. Piensa en un árbol ternario de búsqueda parcial- ArbolTriBP, en el que sus descendencias son *izq* – *med* – *der* una peculiaridad de este árbol es que sus datos estan ordenados parcialmente, es decir que  $izq < med < der$ . Un ejemplo de un árbol ternario de búsqueda parcial se muestra en la figura 4.11 inciso a). Puedes asumir que el árbol nunca genera un nivel a menos que el anterior este lleno. Por otro lado siempre se inserta en el primer lugar que se tiene libre considerando una inserción por amplitud.

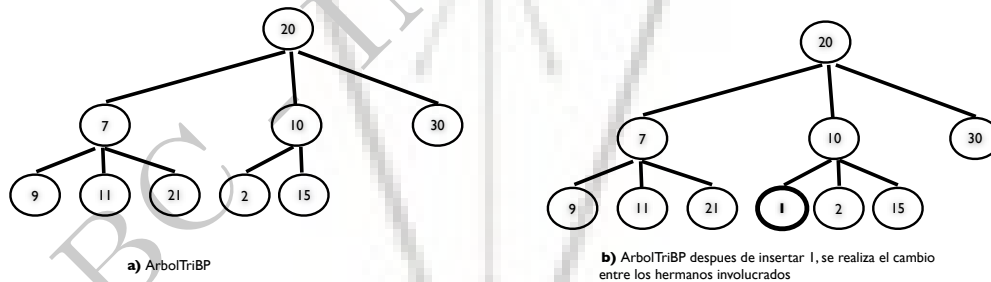


Figura 4.11: Ejemplo

En la figura 4.11, inciso b se muestra como seria el resultado de insertar 1 en la estructura.

Se te pide escribir el método insertar un dato en el ArbolTriBP, líneas abajo se te muestra parte del código de un ArbolTriBP:

```
public class ArbolTriBP<T extends Comparable<T>>
{
 private T raiz;
 private ArbolTriBP<T> izq, med, der;
 public ArbolTriBP(){
 raiz = null;
 izq = med = der = null;
 }
}
```

```

 }
 public void insertar(T d){
 // debes definirlo
 }
}

```

45. El código genético ADN tiene la función de “guardar información”. Es decir, contiene las instrucciones que determinan la forma y características de un organismo y sus funciones. Científicos Biólogos almacenaron muestras de ADN en una estructura de árbol y detectaron muestras de especies desaparecidas, para una mejor organización de su información desean, eliminar de dichas estructuras de almacenamiento todas las muestras que corresponden a estas especies desaparecidas. La información que almacena el ADN está representada por secuencias de caracteres que sólo pueden ser cuatro: A (Adenina), T (Timina), C (Citosina) o G (Guanina). Y el gen que identifica a estas especies desaparecidas son “ATA”, si por ejemplo se tiene la cadena genética “ACAGA” esta no tiene el gen de una especie desaparecida, por lo que se mantiene, pero si fuera “ACATA”, esa cadena tiene el gen de especie desaparecida y debería eliminarse del árbol. Este árbol es muy especial como se ve en la figura 4.12, en este árbol hay 10 cadenas genéticas registradas.

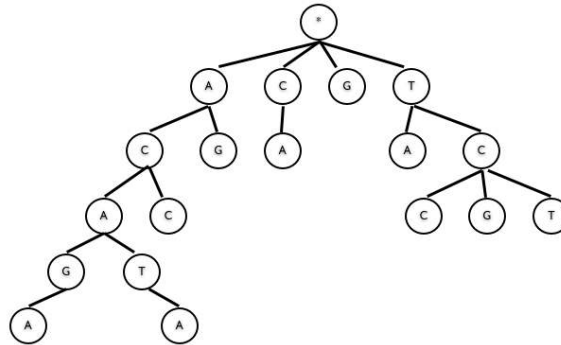
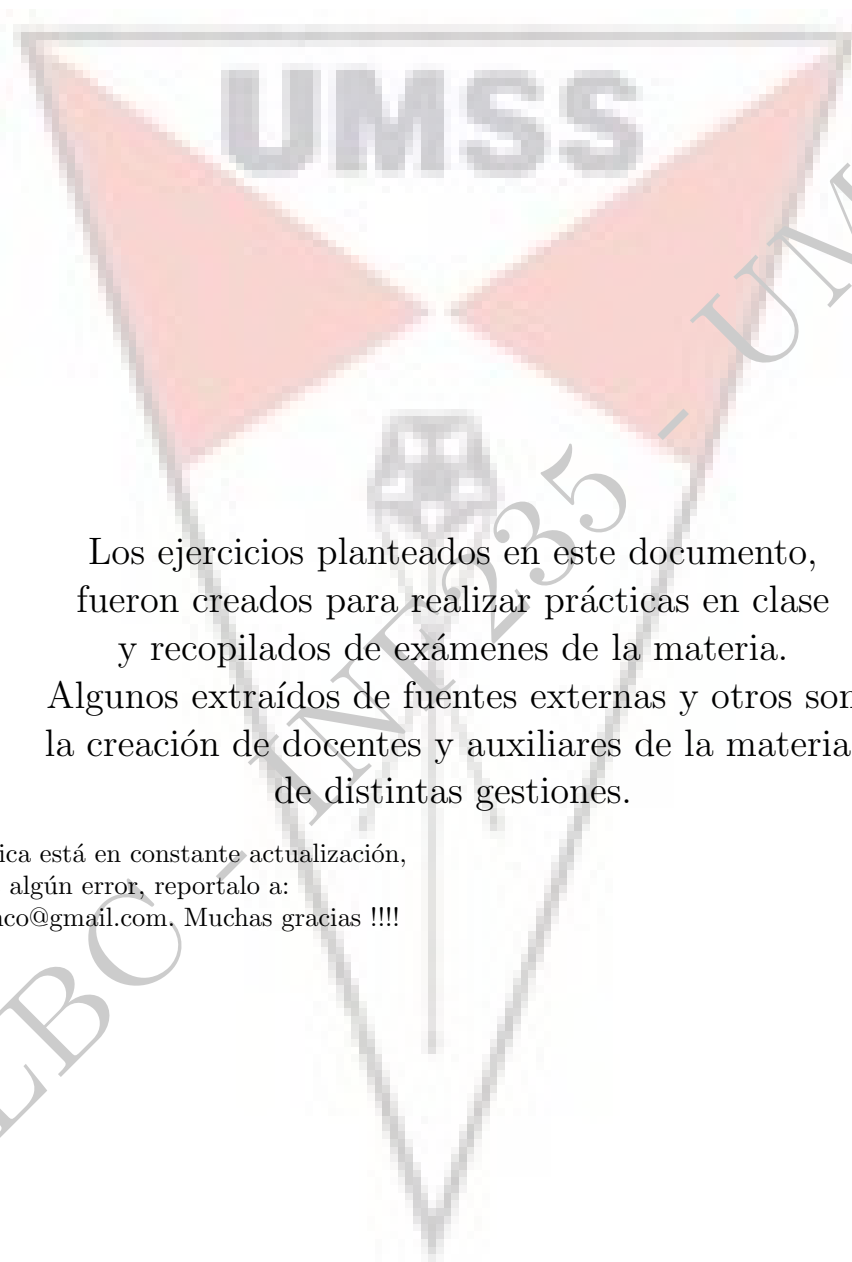


Figura 4.12: Arbol de ADN

- Determina que tipo de árbol requieres para resolver de manera eficiente el problema.
  - Escribe los métodos necesarios que permitan eliminar las muestras de las especies desaparecidas almacenadas. La desaparición debe ser EFICIENTE, mejor si lo haces en una pasada.
46. Un equipo de desarrollo está trabajando en la implementación de un verificador de discursos, de acuerdo a un estudio realizado la mayoría de la población tiene dificultades en la redacción, tienden a repetir una misma palabra varias veces en un párrafo. En una primera etapa se tiene el desafío de dado un párrafo verifique si existe o no palabras repetidas más de dos veces. El reporte debería darlo de manera ordenada con las palabras que se repiten y además de la cantidad de veces que se repiten
- Elija la estructura de datos adecuada para resolver de manera eficiente el problema
  - Describe la estructura de los datos que almacenaran tus estructuras de datos
  - Escribe los métodos necesarios que permitan verificar si el párrafo tiene o no más de dos palabras repetidas.
  - Escribe el métodos que te permita recoger el reporte de las palabras repetidas y sus frecuencias



Los ejercicios planteados en este documento,  
fueron creados para realizar prácticas en clase  
y recopilados de exámenes de la materia.  
Algunos extraídos de fuentes externas y otros son  
la creación de docentes y auxiliares de la materia,  
de distintas gestiones.

Esta práctica está en constante actualización,  
si observas algún error, reportalo a:  
[leticia.blanco@gmail.com](mailto:leticia.blanco@gmail.com). Muchas gracias !!!!