



Programación de Sistemas
Grado en Ingeniería Telemática

Leganés, 4 de mayo de 2018
Duración de la prueba: 20 min

Examen parcial 2 (teoría)
Puntuación: 3 puntos sobre 10 del examen

Sólo una opción es correcta en cada pregunta. Cada respuesta correcta suma 0,3 puntos. Cada respuesta incorrecta resta 0,1 puntos. Las preguntas no contestadas no suman ni restan puntos.

Marca:  Anula:  No uses:   

- Marca la respuesta a cada pregunta con una equis (“X”) en la tabla de abajo.
- Si marcas más de una opción o ninguna opción, la pregunta se considera no contestada.
- Rellena **tus datos personales** antes de comenzar a realizar el examen.

Nombre:

Grupo:

Firma:

NIA:

--	--	--	--	--	--	--	--	--

	A	B	C	D		A	B	C	D
1					6				
2					7				
3					8				
4					9				
5					10				



- 1.- Dada la clase `LinkedList` vista en las sesiones de laboratorio. Indica qué hace el siguiente código:

```
public class Main{

    public static void recursive(LinkedList<Integer> lq){
        Integer i = null;
        if ( !lq.isEmpty() ){
            i = lq.dequeue();
            recursive(lq);
            lq.enqueue(i);
        }
    }

    public static void main(String args []){
        LinkedList<Integer> lq = new LinkedList<Integer>();
        lq.enqueue(new Integer(10)); lq.enqueue(new Integer(15));
        recursive(lq);
    }
}
```

- (a) *** Deja la cola con el mismo número de elementos y orden inverso.
 - (b) Deja la cola vacía.
 - (c) Deja la cola con el mismo número de elementos y en el mismo orden.
 - (d) Deja la cola duplicando los elementos que ya existen.
- 2.- Dado el siguiente código basado en lo explicado en la asignatura de listas dinámicas. ¿Cuál de las siguientes opciones es correcta?

```
public E extract() {
    E data = null;
    data = this.first.getInfo();
    this.first = this.first.getNext();
    return data;
}
```

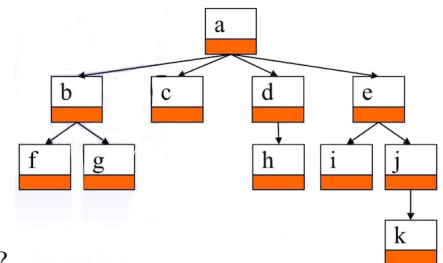
- (a) *** El método da error en alguna ocasión según el estado de la lista.
 - (b) El método siempre devuelve el primer elemento de la lista borrándolo de ésta.
 - (c) El método siempre devuelve el primer elemento de la lista pero no lo borra de ésta.
 - (d) El método no compilaría por el uso incorrecto de *this*.
- 3.- Una pila tiene una política de inserción y borrado de elementos:
- (a) *** LIFO
 - (b) FIFO
 - (c) LILO
 - (d) FILO

4.- Dada la estructura de datos colas dobles (*deque*) si utilizamos el par de métodos *public insertFirst(E info)* y *public e removeFirst()* estaremos usando la cola doble como:

- (a) *** Una pila
- (b) Un árbol
- (c) Una cola
- (d) Ninguna de las anteriores

5.- Si recorres Inorden este árbol de enteros 4, 2, 6, 1, 3, 5, 7 para imprimir por pantalla los elementos que contiene ¿qué resultado se obtendrá?:

- (a) *** Mostrará los números ordenados de menor a mayor.
- (b) Mostrará los números ordenados de mayor a menor.
- (c) Mostrará los números pares primero y luego los números impares.
- (d) Mostrará los números impares primero y luego los números pares.



6.- Dado el siguiente árbol ¿cuál es la altura, profundidad y tamaño?

- (a) *** Altura: 3, Profundidad: 0, Tamaño: 11.
- (b) Altura: 3, Profundidad: 1, Tamaño: 10.
- (c) Altura: 4, Profundidad: 0, Tamaño: 11.
- (d) Altura: 4, Profundidad: 1, Tamaño: 10.

7.- La búsqueda binaria tiene un orden de complejidad de:

- (a) *** $O(\log N)$
- (b) $O(n)$
- (c) $O(n*n)$
- (d) $O(2*\log N)$

8.- En una ordenación de menor a mayor. De estos cuatro algoritmos de ordenación: Bubble Sort, Cocktail Sort, Selection Sort e Insertion Sort ¿Cuál es el que más intercambios hace en un array de muchos elementos?

- (a) *** Bubble Sort
- (b) Cocktail Sort
- (c) Selection Sort
- (d) Insertion Sort

9.- ¿Cuál es la característica que no cumple el siguiente montículo? 11,22,33,20,55,66,77

- (a) *** Para cada nodo (excepto para la raíz) su clave es mayor o igual que la de su padre.
- (b) Completo
- (c) Binario
- (d) Binario de búsqueda

10.- Dada una *LinkedList* y el siguiente método *toString()*. Indique cuáles son los fallos de programación que tiene (nota: si existe nodo entonces siempre responde *getInfo()*):

```
public String toString() {  
    String listText = "";  
    Node<E> current = this.first;  
  
    while(current.getNext() != null) {  
        listText = listText + current.getInfo().toString();  
        current = current.getNext();  
    }  
    return listText;  
}
```

- (a) *** `java.lang.NullPointerException` con lista vacía y no trata el último elemento de la lista.
- (b) `java.lang.NullPointerException` tanto con lista vacía como con lista sólo con un elemento.
- (c) Se pierde la referencia *first* cuando acaba el método.
- (d) Se pierde la referencia *first* cuando acaba el método y no trata el último elemento de la lista.