

Esto no son apuntes pero tiene un 10 asegurado (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)



UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación e Inteligencia Artificial

Estructuras de datos. Curso 2024-2025

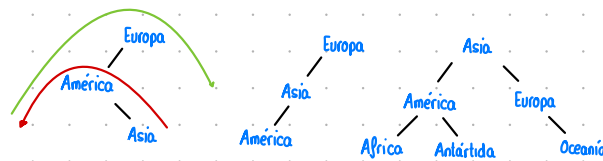
Convocatoria ordinaria de Enero.

Grado en Ingeniería Informática.

Doble Grado en Ingeniería Informática y Matemáticas

Doble Grado en Ingeniería Informática y ADE

1. (1 punto) (a) Si insertamos las cadenas de caracteres {Europa, America, Añía, Antártida, Oceanía y Africa} en ese orden, en un AVL, la etiqueta más a la izquierda de árbol es: (a1) Europa (a2) Africa (a3) Asia (a4) Oceanía (a5) Todo lo anterior es falso. Mostrar el árbol final

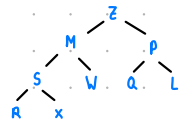


Respuesta: a2

(b) Dados los siguientes recorridos en preorden = (Z,M,S,R,X,W,P,Q,L), y postorden = (R,X,S,W,M,Q,L,P,Z) (b1): No hay ningún árbol binario con esos recorridos asociados; (b2): Hay 1 solo árbol binario con esos recorridos asociados; (b3): Hay dos árboles binarios con esos recorridos asociados; (b4): Todo lo anterior es falso. Razonar la respuesta.

pre = { Z M S R X W P Q L }

post = { R X S W M Q L P Z }



1. raíz
2. P y M
3. descendientes de P y M
4. Q y L
5. W y S
6. descendientes de S
7. R y X

Respuesta: b2

(c) Considerar las siguientes funciones hash (con tamaño de tabla M primo):

- (1) $h(k) = [(k \% M)] + [k \% (M-1)]$ (2) $h(k) = 1 + (k \% M)$ (3) $h(k) = (M * k) \% M$
(4) $h(k) = M \% k$ (5) $h(k) = [k \% (M-1)] \% M$

(c1): Solo una es válida (c2): Solo dos son válidas (c3): Ninguna es válida; (c4): Todas son válidas. Razona la respuesta.

1. No, los valores tienen que ir de 0 a $M-1$, cosa que ocurre en la primera parte, pero al sumar con la segunda parte hay valores que se pasan al hacer la suma. ej. $M=13$ $7 \% 13 + 7 \% 12 = 7 + 7 = 14$ X
 $k=7$
2. No, la expresión $k \% M$ devuelve valores de 0 a $M-1$, pero al sumar 1, los valores van de 1 a M .
3. Correcta, los valores se distribuyen uniformemente de 0 a $M-1$ correctamente.
4. No, los valores se distribuyen según el valor k (de 0 a $k-1$) y k va cambiando, no se distribuye uniformemente y en ocasiones se pasan de las casillas disponibles.
5. No, en primer lugar, hacer un segundo módulo de un número más grande que el primer módulo no cambia nunca el valor, por lo que si nos centramos en $k \% (M-1)$, nunca se ocupará la última casilla de la tabla ($M-1$). Los valores van de 0 a $M-2$.

Respuesta: c1

Consulta condiciones aquí



do your thing

WUOLAH

(d) La función

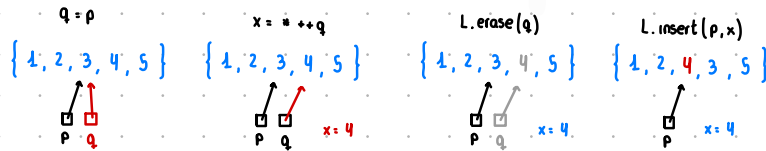
```
void swap(list<int> &L, list<int>::iterator p) {
    int x; list<int>::iterator q;
    /* ... */
}
```

debe intercambiar el elemento en la posición p con el de la posición siguiente (asumiendo que apuntan a elementos válidos de la lista).

¿Cuál de los siguientes es el fragmento correcto que faltaría?

- (d1) q = p; x = *++q; L.erase(q); L.insert(p, x);
 (d2) x = *++p; p = L.erase(p); L.insert(p, x);
 (d3) x = *++p; L.erase(p); L.insert(p, x);
 (d4) x = *p; p = L.erase(p); L.insert(p, x);
 (d5) Ninguno de los anteriores es válido.

Respuesta d1



2. (1 punto) Se desea construir un tasador de seísmos por una magnitud y el nombre de las ciudades en que han ocurrido. Un seísmo con una cierta magnitud puede aparecer en más de una ciudad
- Dar una representación para el TDA tasador usando el tipo `map<float, set<string>>`
 - Implementar la función `insertar` que añade un seísmo dando la magnitud junto con las ciudades en que ha ocurrido.
 - Implementar una función que obtenga las ciudades en que se ha dado un seísmo de determinada magnitud.
 - Implementar la clase `iteradora` dentro de la clase `tasador` para poder iterar sobre todos los seísmos. Han de implementarse (aparte de las de la clase `iteradora`) las funciones `begin()` y `end()`.

```
class tasador {
private:
    map<float, set<string>> datos;

public:
    void insertar (float magnitud, set<string> ciudades) {
        datos[magnitud] = ciudades;
    }

    set<string> obtener_ciudades (float magnitud) const {
        auto it = datos.find(magnitud);
        if (it != datos.end())
            return it->second; // return datos[magnitud];
        else
            return set<string>{};
    }

    class iterator {
private:
        map<float, set<string>>::iterator it;

public:
        iterator() {}
        bool operator == (const iterator &i) const {
            return i.it == it;
        }
        bool operator != (const iterator &i) const {
            return !(*this == i);
        }
    };

    float &operator *() {
        return it->first;
    }
    iterator &operator ++() {
        ++it;
        return *this;
    }
    friend class tasador;
}; //end iterator

iterator begin() {
    iterator i;
    i.it = datos.begin();
    return i;
}

iterator end() {
    iterator i;
    i.it = datos.end();
    return i;
} //end tasador
```

3. (1 punto) Dada una lista de enteros L con elementos repetidos, implementar una función:
list<list<int>> agrupariguales (const list<int> & L);

que construya a partir de ella una lista ordenada de listas, de forma que en la lista resultado los elementos iguales se agrupen en la misma sublista. No puede usarse ningún algoritmo de ordenación, ni la librería algorithm y como estructura auxiliar solo se permite usar una lista

Ejemplo:

Si L={1,3,4,5,6,3,2,1,4,5,5,1,1}, debe devolver LL={ {1,1,1}, {2}, {3,3}, {4,4}, {5,5,5}, {6} }

```
list<list<int>> agrupariguales (const list<int> & L) {
    list<list<int>> lout;
    list<int> aux (L);
    int min, num_veces;

    while (!aux.empty()) {
        buscar_minimo (aux, min, num_veces);

        lout.push_back();
        auto it = lout.end();
        -- it;
        for (int i = 0; i < num_veces; i++)
            (*it).push_back(min);

        aux.remove(min);
    }

    return lout;
}

void buscar_minimo (list<int> & L, int min, int num_veces) {
    min = * (L.begin());
    num_veces = 1;

    for (auto it = L.begin(); it != L.end(); ++it) {
        if (*it < min) {
            min = (*it);
            num_veces = 1;
        }
        else if (*it == min) {
            num_veces++;
        }
    }
}
```

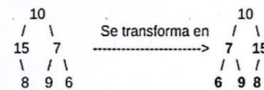
4. (1 punto) Dado un árbol binario de enteros, se dice que está sesgado a la izquierda si para cada nodo, se satisface que la etiqueta de su hijo izquierda es menor que la de su hijo derecha (en caso de tener un sólo hijo, éste ha de situarse necesariamente a la izquierda).

Implementar una función que transforme un árbol binario en un árbol sesgado hacia la izquierda.

void TransformaEnSesgado (bintree<int> a);

La transformación debe preservar que si un nodo v es descendiente de otro w en A, también lo debe ser en el árbol transformado, por lo que no es válido hacer simplemente un intercambio de las etiquetas de los nodos.

Ejemplo:



```
void TransformaEnSesgado (bintree<int> a) {
    TransformaEnSesgado (a, a.root());
}

void TransformaEnSesgado (bintree<int> a, bintree<int>::node n) {
    if (!n.null()) {
        TransformaEnSesgado (a, n.left());
        TransformaEnSesgado (a, n.right());

        if (!n.left().null() && n.right().null()) {
            if (n.left().data() > n.right().data()) {
                bintree<int> a1zq, ader;
                a.prune_left (n, a1zq);
                a.prune_right (n, ader);
                a.insert_left (n, ader);
                a.insert_right (n, a1zq);
            }
        }
        else if (!n.right().null()) {
            bintree<int> aux;
            a.prune_right (n, aux);
            a.insert_left (n, aux);
        }
    }
}
```

Esto no son apuntes pero tiene un 10 asegurado (y lo vas a disfrutar igual).

Abre la Cuenta NoCuenta con el código **WUOLAH10**, haz tu primer pago y llévate 10 €.

Me interesa

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

ING BANK NV se encuentra adherido al Sistema de Garantía de Depósitos Holandés con una garantía de hasta 100.000 euros por depositante. Consulta más información en [ing.es](https://www.ing.es)



5. (1 punto) Dado un `map<T,int> M`, implementar la función:

```
template <class T>
list<T> DimeRangoClave(const map<T,int> &M, const T &s_low, const T &s_up)
```

que dado el map, devuelva las claves en el map que hay entre dos claves, `[s_low, s_up]`. Por ejemplo, si instanciamos `T` a `string`, podríamos llamar a la función para obtener los elementos almacenados entre las entradas `["Patata", "Zanahoria"]`. Un ejemplo de ejecución de la función sería:

```
map<string,int> veg = {{"Anacardo", 3}, {"Zanahoria", 7}, {"Patata", 2},
{"Calabacin", 3}, {"Remolacha", 3}, {"Sandia", 5}, {"Melon", 10},
{"Coliflor", 9}};

string s1 = "Patata", s2 = "Zanahoria";

list<string> l = DimeRangoClave(veg, s1, s2);
// l tendría {"Patata", "Remolacha", "Sandia", "Zanahoria"}
```

```
template <class T>
```

```
list<T> DimeRangoClave (const map<T,int> &M, const T &s_low, const T &s_up) {
    list<T> lout;
    for (auto it = M.begin(); it != M.end(); ++it) {
        if (it->first >= s_low && it->first <= s_up)
            lout.push_back(it->first);
    }
    return lout;
}
```

6. (1 punto) Determinar paso a paso las estructuras resultantes tras:

6-a Insertar las claves {18, 26, 30, 51, 20, 72, 37, 75, 23, 24} en una Tabla Hash cerrada de tamaño 13. A continuación borrar el 23 y el 51 y finalmente insertar el valor 16. Resolver las colisiones usando rehashing doble.

$M = 13$

$h_i(K) = h(K) = K \% 13$

$h_0(K) = 1 + (K \% 11)$

$h_i(K) = (h_{i-1}(K) + h_0(K)) \% M \quad i = 2, 3, \dots$

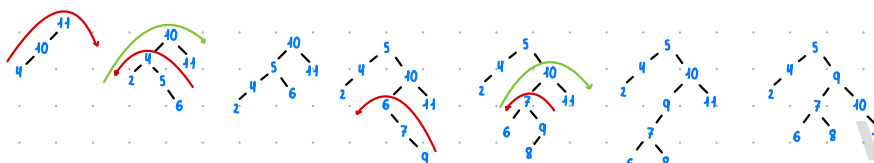
- $h(18) = 5$
- $h(26) = 0$
- $h(30) = 4$
- $h(51) = 12$
- $h(20) = 7$
- $h(72) = 7$
- $h(37) = 11$
- $h(75) = 10$
- $h(23) = 10$
- $h_0(23) = 2$
- $h_2(23) = 12$
- $h_3(23) = 1$
- $h_4(23) = 3$
- $h(24) = 11$
- $h_0(24) = 3$
- $h_2(24) = 1$
- $h_3(24) = 4$
- $h_4(24) = 7$
- $h_5(24) = 10$
- $h_6(24) = 0$
- $h_7(24) = 3$
- $h_8(24) = 6$
- $h(16) = 3$

	K	dir	estado
0	26		x
1	72		x
2			
3	16		x
4	30		x
5	18		x
6	24		x
7	20		x
8			
9			
10	75		x
11	37		x
12	51		x

6-b Insertar en el orden indicado (detallando los pasos) las siguientes claves en un AVL<int>:

{11, 10, 4, 5, 2, 6, 7, 8}

{11, 10, 4, 5, 2, 6, 7, 8}



Consulta condiciones aquí



do your thing