

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)



TEMA 1: Eficiencia de algoritmos.

Principio de invarianza: Dos implementaciones diferentes de un mismo algoritmo no difieren mas que en k.

Eficiencia: Medida de uso de recursos en funcion del tamaño de entrada.

NOTACIÓN $O(n)$

- ° $f(n)$ es $O(f(n))$ si existen c cte y n_0 tal que $f(n) < c \cdot f(n)$ siendo $n > n_0$.
- ° Para una función elemental $\rightarrow O(1)$.

NOTACIÓN $\Omega(n)$

- ° $f(n)$ es $\Omega(f(n))$ si existen c cte y n_0 tal que $f(n) > c \cdot f(n)$ siendo $n > n_0$.

NOTACIÓN $\Theta(n)$

- ° Se da cuando $f(n)$ es $O(n) = \Omega(n) \rightarrow f(n)$ es $\Theta(n)$.

BUCLES

Un bucle se realiza n veces y se calcula : $n-s-v+1$ siendo s el simbolo ($<$ o \leq) y v el valor inicial.

→ **Condicionales:**

- ° Sera el maximo tiempo de la parte if y de la del else

→ **Bloque sentencia:**

- ° Se aplica la regla suma

→ **Llamadas a funciones:**

- ° Medida del tamaño de los argumentos.

Consulta condiciones aquí



do your thing

WUOLAH

TEMA 2: Abstracción de datos

Abstracción

~Abstracción en programación

Abstracción: Operación en la cual se ignora para mejorar su comprensión.

Abstracción en la resolución de problemas: Ignorar detalles específicos buscando generalidades para obtener una perspectiva distinta.

Descomposición: se varia el nivel de detalle.

- Todas las partes al mismo nivel.
- Cada parte se puede abordar por separado.
- Las partes debe poder unirse en una solución final

~Mecanismos de abstracción en programación

Abstracción por parametrización: se introducen parámetros.

Ej: `sqrt(valor)`

Abstracción por especificación: descripción precisa de su comportamiento

Ej: Requisitos:

Efecto:

→ **Abstracción por especificación:**

-**Precondiciones:** Condiciones necesarias para que el programa funcione correctamente.

-**Postcondiciones:** Enunciados que se suponen ciertos tras la ejecución del programa.

~Tipos de abstracción en programación

Abstracción procedimental: conjunto de operaciones que se comparten como una sola.

Abstracción de datos: conjuntos de datos y operaciones que caracterizan el comportamiento de los datos.

Abstracción de iteración: Trabajar sobre colecciones sin preocuparse por la forma en la que se organizan.

→ **Abstracción procedimental:**

- Realiza operaciones sobre un conjunto de datos. La identidad de estos es irrelevante.

- Importa el que se hace , no el cómo.

-Localidad: no es necesario conocer la implementación de otras abstracciones.

-Modificabilidad: podemos cambiar la implementación sin que afecte a otras , mientras no se modifique la implementación.

Especificación

- Todo debe de ir entre `/** */`

Cabecera: nombre del programa y datos varios (@brief)

Cuerpo:

1. Parámetros____(@param)
2. Requisitos ____(@pre)
3. Valores retorno_(@return)
4. Efectos_____(@post)

Tipos de datos abstractos

Especificación: descripción del comportamiento del TDA.

Representación: Forma en la que se representan los datos.

Implementación: Forma en la que se representan las operaciones.

~Visiones

Visión externa: especificación.

Visión interna: representación e implementación.

~Función de abstracción

- Define el significado de un objeto rep de cara a un objeto abstracto.

- Establece una función de relación formal entre objeto rep y un objeto abstracto.

Ej: {dd,mm,aa} → dd/mm/aa

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)

→ Invariante de la representación:

° Condiciones lógicas que indican si un objeto rep es válido en tipo abstracto.

Ej: fecha tiene día mes y año válidos.

Consulta
condiciones aquí



do your thing

WUOLAH

TEMA 3 : Contenedores básicos.

Contenedores

Contenedores: Estructuras que almacenan datos de un mismo tipo base. Ofrecen métodos para gestionarlos.

Pilas

Tipo de estructura de datos lineal con un comportamiento LIFO , el extremo de la pila (donde se realizan funciones) se llama tope.

~Métodos

`push(x)`: añade un elemento a la pila.

`pop()`: elimina el tope de la pila.

`top()`: devuelve el tope de la pila.

`empty()`: devuelve si la pila está vacía o no.

`size()`: devuelve el tamaño de la pila.

Colas

Estructura de datos lineal con un comportamiento FIFO. El elemento sobre se llama frente.

~Métodos

`push(x)`: añade un elemento a la pila.

`pop()`: elimina el tope de la pila.

`front()`: devuelve el tope de la pila.

`empty()`: devuelve si la pila está vacía o no.

`size()`: devuelve el tamaño de la pila.

~Colas con prioridad

Las inserciones se hacen según una prioridad. Debe tener definido el operador <.

→Métodos

`push(x)`: añade un elemento a la pila.

`pop()`: elimina el tope de la pila.

`front()`: devuelve el tope de la pila.

`empty()`: devuelve si la pila está vacía o no.

`size()`: devuelve el tamaño de la pila.

→Prioridad

- Podemos cambiar la prioridad de la siguiente manera:

`{priority_queue<tipo_dato>,{secuencia},{comparado}>}`

Pair

- Una clase muy simple que permite usar pares de valores.

~Métodos

`first`: devuelve el primer elemento del pair.

`second`: devuelve el segundo elemento del pair.

`x = y`: operador de asignación,

`x < y`: operador de comparación

Tipos de contenedores asociativos

→`set<T>`: Solo almacena valores de clave. No hay valores repetidos

→`multiset<T>`: Como set pero permite valores repetidos.

→`map<key,T>`: Almacena pares (clave , dato) pero no puede haber claves repetidas.

→`multimap<key,T>`: Como map pero permite valores repetidos en la clave.

~Métodos

`insert(x)`: inserta el valor en el contenedor.

`erase(iter)`: elimina el valor dado por el iterador.

`erase(x)`: elimina el elemento x.

`find(x)`: devuelve un iterador al elemento x, sino esta devuelve `end()`.

`count(x)`: devuelve el numero de veces que aparece x.

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)

Contenedores secuencia

→**vector<T>**: se acceden a los elementos por el índice de la posición que ocupan. Secuencia más simple. Tamaño variable.

→**list<T>**: cuando se quieren hacer inserciones y borrados en el medio. Mas eficiente que los vectores.

→**deque<T>**: mayoría de inserciones y borrados al principio y al final.

Consulta
condiciones aquí



do your thing

WUOLAH

TEMA 4 : Contenedores complejos.

Árboles binario

Un árbol de tipo T es un conjunto de nodos T que almacenan elementos mediante una relación padre-hijo.

- Un nodo es un árbol vacío.
- Un árbol vacío (0 nodos) también es un árbol.
- Si hacemos que la raíz de un árbol sea hijo de un nodo de otro árbol también obtenemos un árbol.

~Terminología

- El nodo por encima es el padre, y el de abajo el hijo.
- Nodo raíz es el único que no tiene padre.
- Nodo hoja es aquel que no tiene hijos.
- La información que contiene el nodo se llama etiqueta.
- La profundidad de un nodo es los padres que tiene hasta llegar al nodo raíz.
- La altura es el número de relaciones hijo hasta alcanzar su nodo hoja más lejano.
- Nivel es el conjunto de nodos a la misma profundidad
- Árbol binario es aquel que solo tiene un máximo de 2 hijos por nodo.

~Métodos

→Árbol:

`root()`: devuelve la raíz del árbol.

`size()`: devuelve la cantidad de nodos del árbol.

→Node:

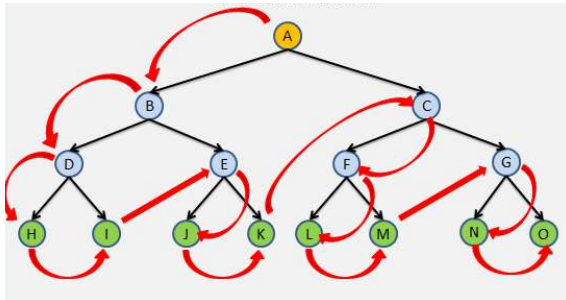
`parent()`: devuelve el padre del nodo.

`left()` / `right()`: devuelve el hijo del nodo.

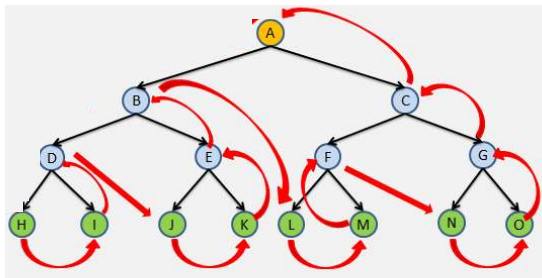
`*nodo`: devuelve la etiqueta del nodo.

~Recorrer el árbol

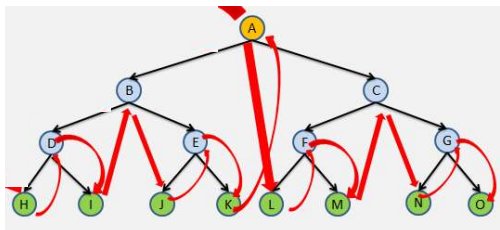
→ **Preorden:** visitar primero la raíz, luego hijo izquierdo y luego el derecho.



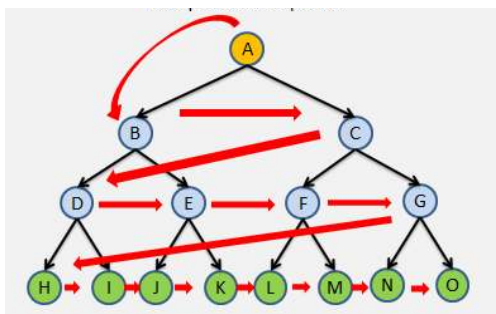
→ **Postorden:** visitar el hijo izquierdo, luego el derecho y luego la raíz.



→ **Inorden:** se visita el hijo izquierdo luego la raíz y luego hijo derecho



→ **Niveles:** se recorre los nodos de izquierda a derecha siguiendo las profundidades.



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)



~Tipos

→Binario de búsqueda(BST):

Definición

- Proceso rápido de búsqueda $O(\log n)$.
- Las etiquetas tienen relación de orden.
- La etiqueta de los descendientes por la izquierda son menores que la etiqueta.
- La etiqueta de los descendientes por la derecha son mayores que la etiqueta.

Propiedades

- Para cada nodo, dada una etiqueta, si la comparamos con la de un nodo sabremos en qué subárbol buscar.
- No basta con que para cada nodo el hijo de la izquierda sea menor y el de la derecha mayor. Un nodo debe de ser coherente con su nodo abuelo.
- Tipo de dato orientado a la búsqueda.
- Operaciones con eficiencia de $O(\log n)$.
- Recorrido en inorden da el conjunto de etiquetas ordenado.

Borrado de elementos

- Si es hoja no pasa nada.
- Si tiene un solo hijo se elimina ese nodo y el hijo ocupa el lugar del padre.
- Si tiene hijos con más hijos se busca un nodo que lo pueda sustituir.

→Binario de búsqueda equilibrado(AVL):

- Proceso rápido de búsqueda $O(\log n)$.
- Número de comprobaciones es independiente del camino que se siga.

Consulta condiciones aquí



do your thing

WUOLAH

- Para cada nodo se cumple que, la diferencia de altura de sus hijos es ≤ 1 y tiene un orden igual que los BST.

- Varía en la implementación de insertar y borrar.

- No puede haber elementos repetidos.

- Se borra igual que un BST.

→Binario parcialmente ordenado(POT):

- Árboles equilibrados con $O(\log n)$ para obtener el mínimo o máximo.

- Cada nodo \leq que sus 2 hijos.

- Para la inserción:

- Si su último nivel no está completo: se inserta lo más a la izquierda posible.

- Si está completo se crea nivel y se añade lo más a la izquierda posible.

- En el borrado solo se puede eliminar la raíz, el elemento del último nivel más a la derecha pasa a ser la nueva raíz.

Tablas hash

- Diseñado para búsquedas ideales $O(1)$ aplicando una función matemática.

- Si se obtienen valores diferentes para cada clave se obtiene la función hash perfecta.

- Dos claves distintas con el mismo valor produce una colisión.

~Características

- Consistentes: para la misma clave se debe obtener siempre el mismo valor.

- Debe tomar como valor cada uno de los valores de la tabla.

- Debe de ser $O(1)$.

- Debe de minimizar el número de colisiones.

~Función hash

→Metodos:

Multiplicacion

- Multiplicar por un valor y luego seleccionar alguno de los bits de la expresión binaria.

Division

- Se calcula el modulo a M , siendo M el tamaño y primo.

Mixto

- $(ax+b)\%M$ siendo $a\%M \neq 0$

→Dato:

Entero

- Podemos usar el mixto.

String

- Algoritmos djb2 o sdbm.

Dato compuesto

- Se puede obtener en función de los valores hash de los componentes.

~Resolución de colisiones

→Encadenamiento separado(hash abierto):

- Se soluciona insertando los elementos colisionados en una lista, de esta manera se obtiene un vector de listas.

→Direccionamiento abierto(hash cerrado):

- Se soluciona calculando valores hash hasta que no se produzcan colisiones y se encuentre un hueco.

Reasignación de casilla

- Prueba lineal: $h_i(x) = (h(x) + i) \% M$
- Prueba cuadrática: $h_i(x) = h(x) + i^2$
- Hashing doble: $\{h_i(x) = (h(x) + i * h'(x)) \% M\} \Rightarrow h'(x) \neq 0 \ \&\& \ h'(x) = q - (x \% q)$ siendo q un primo $< M$

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)



Operaciones

- Las búsquedas siguen la misma secuencia que para la inserción.
- Los borrados dejan las casillas en un estado diferente a Libre u Ocupado.
- Las casillas borradas se consideran libres para la inserción y ocupadas para la búsqueda.
- Si la tabla se llena se hace un rehashing.

Grafos

- Se dice dirigido cuando el orden de los componentes es relevante $(x,y) \neq (y,x)$. Sus lados se denominan arcos.
- No se dice dirigido cuando el orden de los componentes es irrelevante. Sus lados se denominan aristas.

→Conceptos:

- Camino entre u y v si hay pareja de pares tal que : $(u,v_1), (v_1,v_2), \dots, (v_n,v)$.
- Longitud del camino es el número de pares.
- Un camino es ciclo si $u=v$.
- Casos importantes:
 - Grafos dirigidos acíclicos.
 - Árboles.

→Recorridos:

Profundidad

- Numeración en preorden, se le asigna a cada vértice un número de orden según cómo se visitan. Añadir al principio del recorrido.

Consulta condiciones aquí



do your thing

WUOLAH

- Numeración en postorden, se le asigna a cada vértice un número después de haber visitado todos sus nodos adyacentes. Añadir al final del recorrido.

Anchura

- La idea es visitar un nodo y luego a un vecino y así sucesivamente.
- Se usa para grafos infinitos o encontrar un camino.

B-árboles

- Un b-árbol de orden m cumple:
 - $1 \leq \#(\text{claves en la raíz}) \leq 2m$
 - $m \leq \#(\text{claves en el nodo}) \leq 2m$
 - Todos los nodos hoja tienen la misma profundidad
 - Cada nodo interno con s claves tiene $s+1$ hijos
- Los árboles 2-3 son un caso de b-árboles con $m=1$.

→ Borrados:

- Se eliminan siempre de una hoja.

Se cumple el mínimo de hojas

- Se deja como esta, ya que no hay nada que cambiar y cumple los requisitos.

No está en una hoja

- Se elimina al sucesor/predecesor que se encuentre en una hoja.

Se quedan menos claves que el mínimo

- Se miran las hojas adyacentes:
 - Si una tiene más del mínimo una de sus claves pasa al padre y bajamos una del padre a la hoja.
 - Si ninguna tiene más del mínimo esa hoja y una de las vecinas se combinan con el padre.