Estructuras de Datos
Curso 2016/17

Datos Académicos

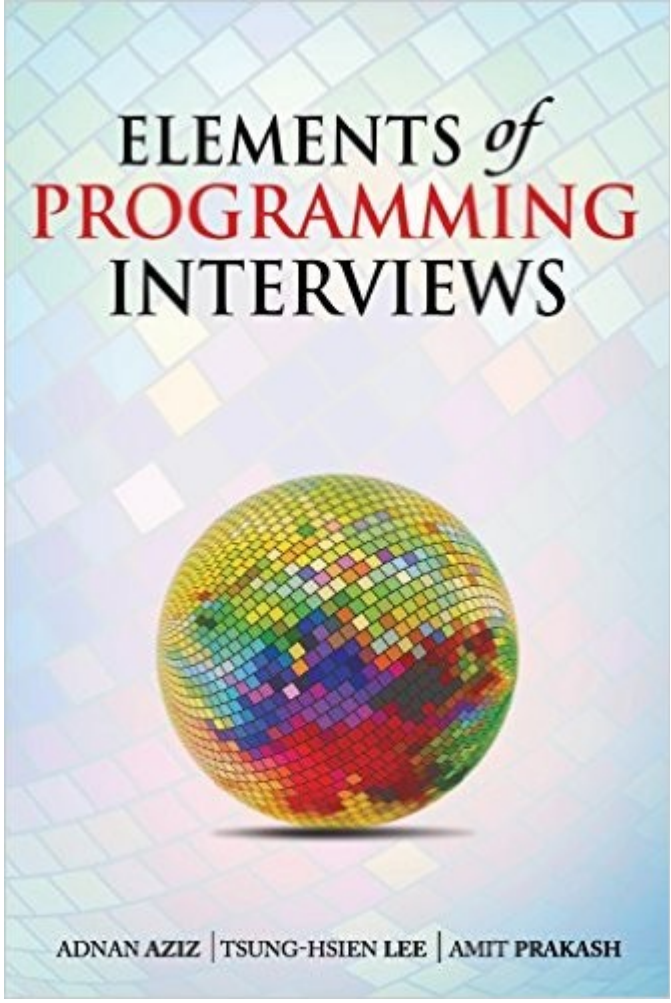
- Dpto. Ciencias de la Computación e I.A.
- Profesor Teoría / Prácticas
  - Juan F. Huete
  - Email: jhg@decsai.ugr.es
  - Despacho D21- 4 Planta
- Página Web http://decsai.ugr.es/

Tutorías L (8:30 a 9:30) M (9:30 a 10:30) y L,M,X,J (13:30 a 14:30)

- Comienzo Grupo de Prácticas: Próxima Semana
  - OBLIGATORIO DIVISION DE ALUMNOS (MAX 30)
  - LUNES, MARTES y VIERNES 11:30 a 12:30

- “Developing problem solving skills is like learning to play musical instruments. -- books and teachers can point you in the right direction, but only your hard work will take you there. Just as a musician, you need to know underlying concepts, but theory is not substitute for practice.”

- Elementos a utilizar ...
  - Estructuras de Datos
  - Diseño de algoritmos
  - Patrones de Diseño Abstracto

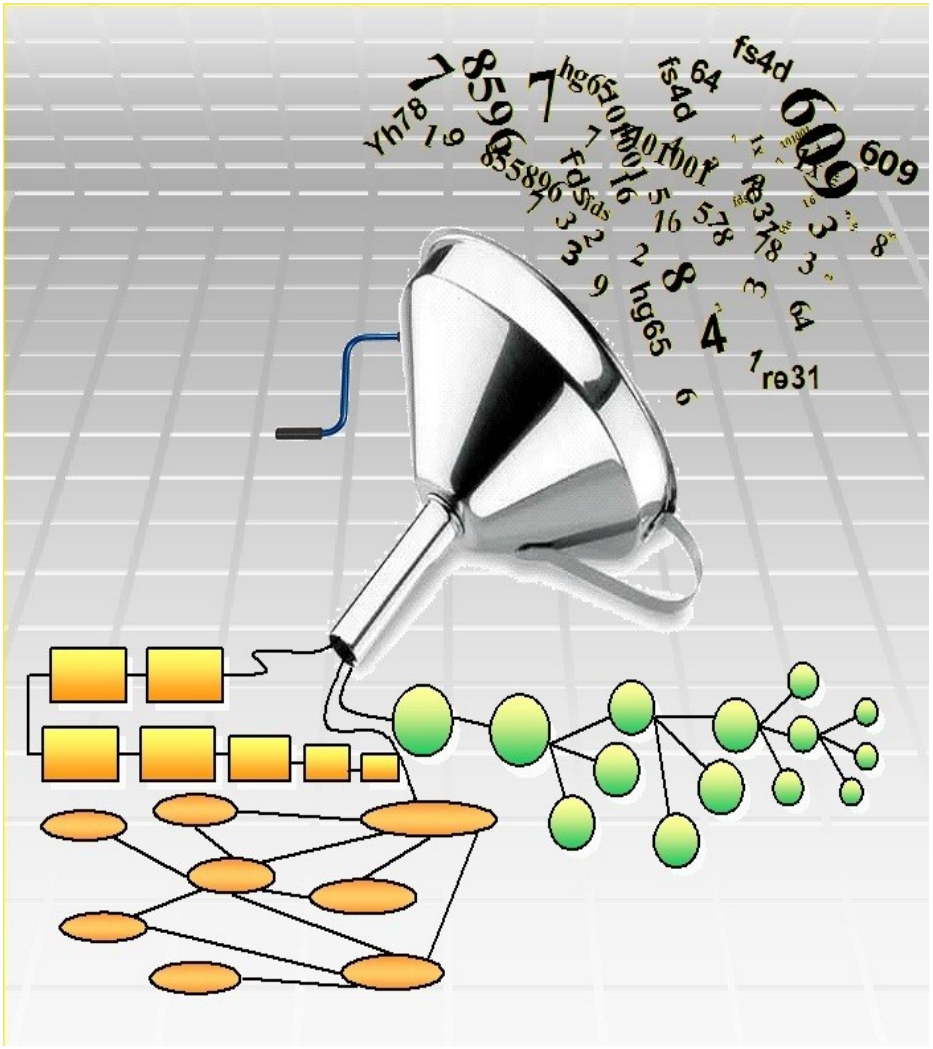


Estructuras de Datos

- Estrategias para almacenar y organizar elementos relacionados de forma que se puedan manipular de forma eficiente.

- La elección de una buena estructura de datos es la llave para el diseño de un buen algoritmo.

- Normalmente, para solucionar un problema se requiere del uso de varias estructuras distintas.



Ejemplos:

Gestionar:

- Guía de teléfonos: asociar a un nombre un teléfono
- DNS: busca por el nombre del sitio web para encontrar su dirección IP.
- Sistema de Ficheros: buscar por un nombre para encontrar su dirección en el disco duro.
- Gestión de alumnos: buscar por DNI para encontrar las asignaturas matriculadas.
- Indice de un libro: determinar en qué página aparece cada palabra
- ....

¿ Qué tienen en común estos problemas ?

Ejemplos:

Gestionar:

- Guía de teléfonos: buscar por un nombre para encontrar el teléfono
- DNS: busca por el nombre del sitio web para encontrar su dirección IP.
- Sistema de Ficheros: buscar por un nombre para encontrar su dirección en el disco duro.
- Gestión de alumnos: buscar por DNI para encontrar las asignaturas matriculadas.
- Indice de un libro: determinar en qué página aparece cada palabra
- ....

Necesitamos de mecanismos para

Insert: inserta el par nombre-dominio::dirección IP.

Get: busca por el dominio y devuelve la dirección IP.

Erase: borrar un nombre de dominio

Size: devuelve el numero de nombres de dominio

Copy: hacer una copia

.....

Debemos permitir buscar, insertar, acceder en cualquier orden sobre cualquier colección de datos

Objetivo: Seleccionar la implementación más eficiente.

- Utilizar una estructura u otra puede hacer que un algoritmo se ejecute en segundos o varios días

Objetivos I

Reconocer la importancia de la abstracción y conocer los tipos de abstracciones que aparecen en programación: funcional, de datos, de iteración y abstracción por generalización.

Saber diferenciar entre la especificación, representación e implementación de un tipo de dato abstracto, conociendo los conceptos de Función de Abstracción e Invariante de la Representación.

Comprender cómo los conceptos de ocultamiento de información y encapsulamiento ayudan al desarrollo de tipos de datos más fiables.

Comprender los métodos de especificación: basados en una definición mediante axiomas o el método constructivo u operacional (basado en el uso de precondiciones y postcondiciones).

Ser capaz de diseñar e implementar pequeñas aplicaciones para cada uno de los distintos tipos de datos que se impartan en la materia (listas, pilas, colas, colas con prioridad, conjuntos, diccionarios, árboles, tablas hash, grafos).

Objetivos II

Adquirir la capacidad para comprender cómo el uso de distintos tipos de datos afecta a la eficiencia de los algoritmos que la usan.

Ser capaz de implementar en lenguajes de alto nivel los tipos de datos propios de la materia así como otros definidos por el usuario.

Conocer las distintas representaciones e implementaciones de los tipos de datos que se imparten en la materia.

Ser capaz de comparar implementaciones alternativas para un tipo de dato analizando los factores que influyen en la eficiencia y el uso de memoria.

Adquirir la capacidad de evaluar las necesidades de una aplicación específica, tomando decisiones justificadas sobre los tipos de datos y la representación más adecuadas.

Temario

- 1.- Introducción a la eficiencia de algoritmos.
- 2.- Abstracción de datos
- 3.- TDAs contenedores básicos.
- 4.-TDAs contenedores complejos.

- Lo detallaremos más adelante ….

Evaluación

- Evaluación Única
- Evaluación Continua
  - Febrero
  - Septiembre

Evaluación Única Final

- Se realiza en un solo acto académico
- El estudiante, en las dos primeras semanas de impartición de la asignatura, lo debe solicitar al Director del Departamento, alegando y acreditando las razones que le asisten para no poder seguir el sistema de evaluación continua.

Evaluación Continua

- Febrero:
  - Teórica (60):

    Un examen al final del cuatrimestre.
  - Práctica (30):

    Evaluación durante el curso

    Prácticas a realizar a lo largo del cuatrimestre

    (2 prácticas puntuables).
  - Otros (10):

    Participación ACTIVA del alumno

Evaluación Continua

- Septiembre:
  - Examen final con preguntas teóricas y prácticas con un valor de 10 puntos.
  - El alumno podrá guardar la nota obtenida en la convocatoria de Junio en los bloques de "Parte Práctica" y en ese caso la calificación final de la parte teórica se ajustará a 7 puntos

Originalidad de los trabajos y pruebas

- 2. El plagio, entendido como la presentación de un trabajo u obra hecho por otra persona como propio o la copia de textos sin citar su procedencia y dándolos como de elaboración propia, conllevará automáticamente la calificación numérica de cero en la asignatura en la que se hubiera detectado, independientemente del resto de las calificaciones que el estudiante hubiera obtenido. Esta consecuencia debe entenderse sin perjuicio de las responsabilidades disciplinarias en las que pudieran incurrir los estudiantes que plagien.
- 3. Los trabajos y materiales entregados por parte de los estudiantes tendrán que ir firmados con una declaración explícita en la que se asume la originalidad del trabajo, entendida en el sentido de que no ha utilizado fuentes sin citarlas debidamente.

Bibliografía

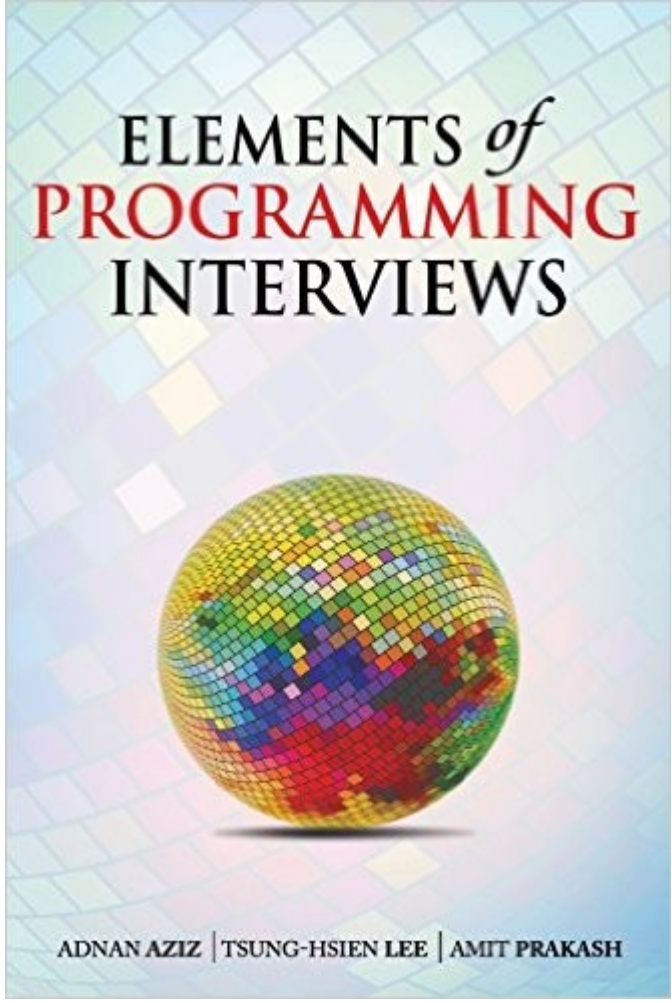
- H.Deitel and P. Deitel. C++ How to program (Early Objects Version) 9/E (2013) Pearson
- Robert Robson Using the STL: The C++ Standard Template Library .Springer; Edición: 2.2013
- Budd, T. (1998). Data structures in C++ using the STL. Addison-Wesley.
- R. Musser, J. Derge y A. Saini. (2009) STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library. 3 Edicicion. Adisson-Wesley 2009.
- Liskov, B. Program Development in Java (2001) Addison-Wesley
- Garrido A, Fernández J. Abstracción y Estructuras de datos en C++. Delta Publicaciones (2006)
- C++ Reference (en inglés) http://www.cppreference.com
- • C Plus Plus (en inglés) http://www.cplusplus.com

Me será útil en el futuro ?

- A nivel academico
  - Algorítmica
  - Resto de asignaturas ...
- A nivel profesional
  - Esencial para trabajar como programador en grandes compañías,
    - Facebook,
    - Amazon,
    - Google,
    - Yahoo,
    - Apple,

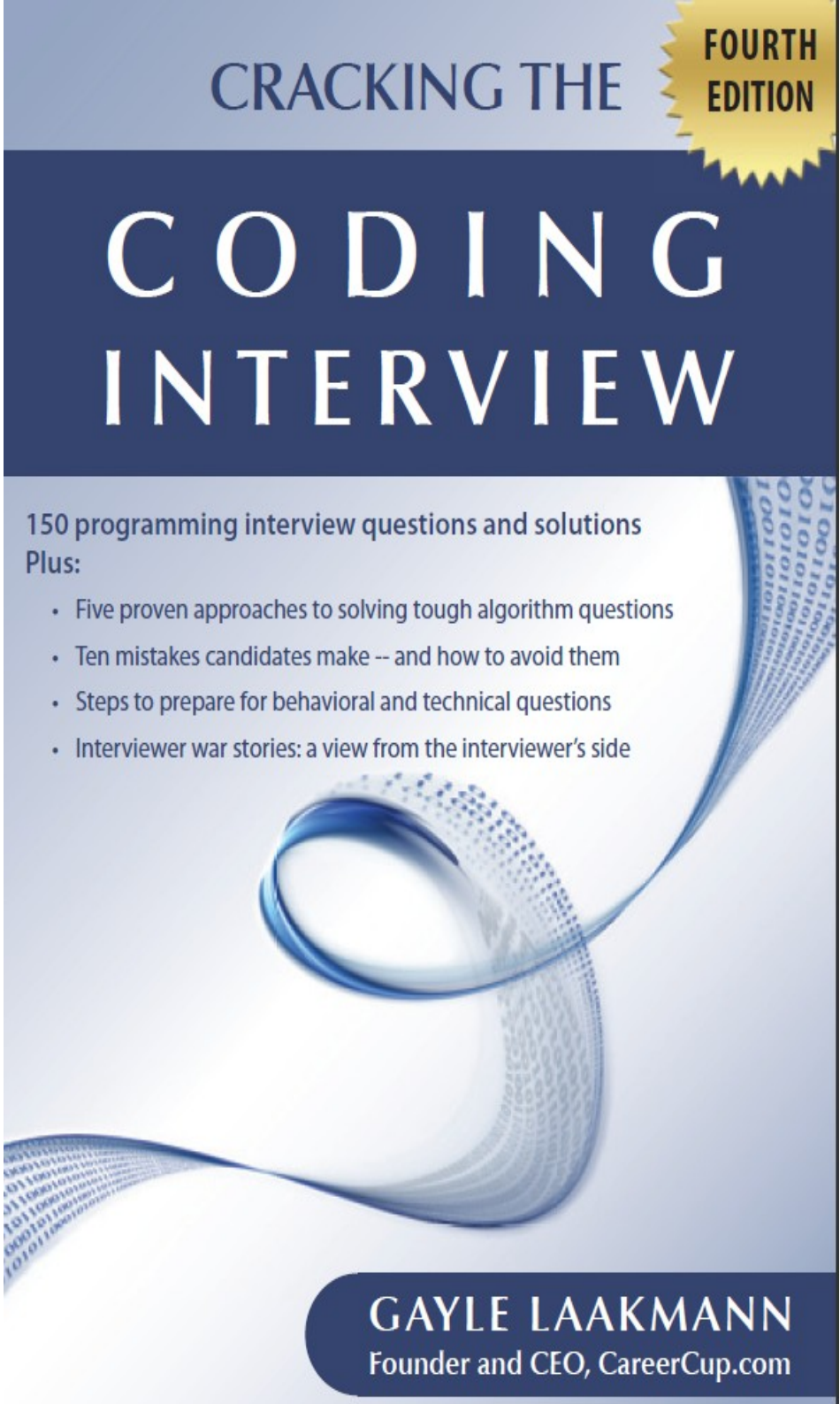
Table 4.1: Data structure patterns.

Data structure	Key points
Primitive types	Know how `int`, `char`, `double`, etc. are represented in memory and the primitive operations on them.
Arrays	Fast access for element at an index, slow lookups (unless sorted) and insertions. Be comfortable with notions of iteration, resizing, partitioning, merging, etc.
Strings	Know how strings are represented in memory. Understand basic operators such as comparison, copying, matching, joining, splitting, etc.
Lists	Understand trade-offs with respect to arrays. Be comfortable with iteration, insertion, and deletion within singly and doubly linked lists. Know how to implement a list with dynamic allocation, and with arrays.
Stacks and queues	Understand insertion and deletion. Know array and linked list implementations.
Binary trees	Use for representing hierarchical data. Know about depth, height, leaves, search path, traversal sequences, successor/predecessor operations.
Heaps	Key benefit: $O(1)$ lookup find-max, $O(\log n)$ insertion, and $O(\log n)$ deletion of max. Node and array representations. Min-heap variant.
Hash tables	Key benefit: $O(1)$ insertions, deletions and lookups. Key disadvantages: not suitable for order-related queries; need for resizing; poor worst-case performance. Understand implementation using array of buckets and collision chains. Know hash functions for integers, strings, objects. Understand importance of equals function. Variants such as Bloom filters.
Binary search trees	Key benefit: $O(\log n)$ insertions, deletions, lookups, find-min, find-max, successor, predecessor when tree is balanced. Understand node fields, pointer implementation. Be familiar with notion of balance, and operations maintaining balance. Know how to augment a binary search tree, e.g., interval trees and dynamic k-th largest.



List of the absolute, must-have knowledge

- Data Structures
  - Linked Lists, Binary Trees, Tries, Stacks, Queues, Vectors, Hash Tables
- Algorithms
  - Breadth First Search, Depth First Search, Binary Search, Merge Sort, Quicksort, Tree Insert/ Find /e.t.c
- Concepts
  - Bit Manipulation, Singleton Design Pattern, Factory Design Pattern, Memory (Stack vs. Heap), Recursion, Big-OTime



Ejemplo: DNS (Servidor de Nombres de Dominio)

- Objetivo: cometido es buscar a partir del nombre de un ordenador (inteligible y fácil de recordar) la dirección IP de ese ordenador; y viceversa, encontrar su nombre a partir de la dirección IP.

Sitio Web	Dirección IP
decsai.ugr.es	150.214.191.180
www.ugr.es	150.214.20.1
www.princeton.edu	128.112.132.86
www.wikipedia.es	93.93.112.62

Algunos datos.
Fuente: INFORME SOBRE LA INDUSTRIA DE NOMBRES DE DOMINIO Sep. 2015 (verisign.com)

Domain Name Industry Brief

The second quarter of 2015 closed with a base of 296 million domain name registrations across all top-level domains (TLDs), an increase of 2.2 million domain names, or 0.8 percent over the first quarter of 2015. Registrations have grown by 16.4 million, or 5.9 percent, year over year. [1]

[1] The generic top-level domain (gTLD) and ccTLD data cited in this report are estimates as of the time this report was developed, and is subject to change as more complete data is received. Total includes ccTLD Internationalized Domain Names.

Algunos datos...

DNS Query Load

During the second quarter of 2015, Verisign's average daily Domain Name System (DNS) query load was 111 billion across all TLDs operated by Verisign, with a peak of 182 billion.

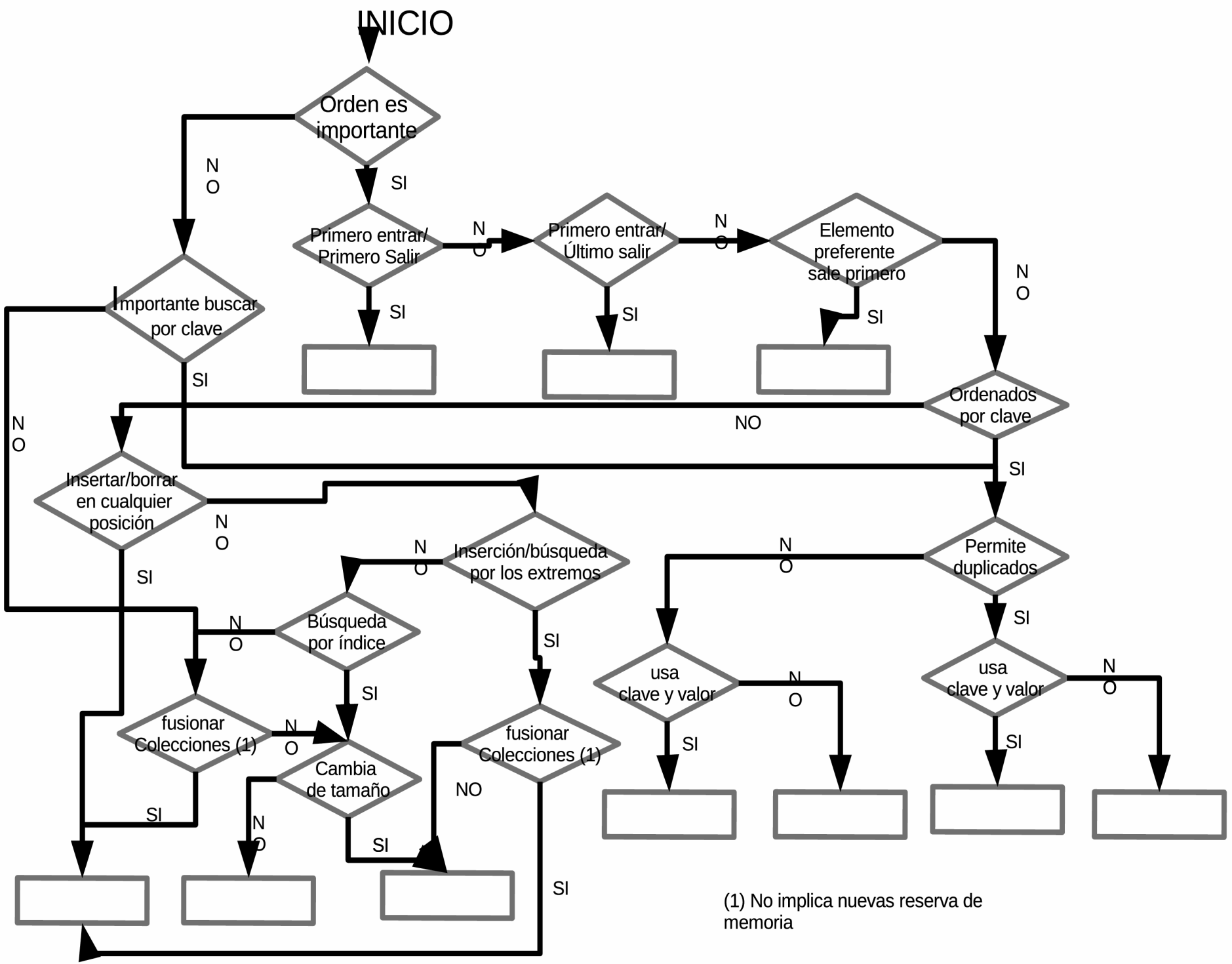
Rank	Domain	Registered	Designation
1	.COM	85336063	Commercial
2	.DE	13459604	Germany / Deutschland
3	.NET	12631270	Network
4	.CO.UK	3080659	UK Commercial
5	.ORG	8333855	Organization
6	.INFO	5828223	Information
7	.IT	3681779	Italy
8	.BIZ	2666399	Business
9	.NL	617045	The Netherlands
10	.CC	581147	Cocos (Keeling) Islands
11	.TV	473168	Tuvalu
40	.COM.HK	200086	Hong Kong Commercial
41	.SK	149644	Slovakia
42	.ES	136539	Spain
Total		136.061.532	

GESTION DEL DNS

- Podemos encontrar diferentes alternativas (implementaciones) y cada una de ellas tiene un comportamiento diferente.

- Ejemplos:
  - ????

Objetivo: Seleccionar la implementación más eficiente.


INICIO
Orden es importante
NO
SI
Primero entrar/ Primero Salir
NO
Primero entrar/ Último salir
NO
Elemento preferente sale primero
NO
SI
SI
SI
SI
Importante buscar por clave
SI
NO
Ordenados por clave
NO
SI
Insertar/borrar en cualquier posición
NO
SI
Inserción/búsqueda por los extremos
NO
Búsqueda por índice
NO
SI
SI
fusionar Colecciones (1)
NO
Cambia de tamaño
NO
SI
SI
NO
fusionar Colecciones (1)
NO
SI
Permite duplicados
NO
SI
usa clave y valor
NO
SI
usa clave y valor
NO
SI
(1) No implica nuevas reserva de memoria

Módulo 1.- Introducción a la eficiencia de algoritmos

- Eficiencia y complejidad en tiempo y espacio.
- Cotas de eficiencia.
- Cálculo del tiempo de ejecución de un algoritmo.

Módulo 2.- Abstracción de datos

- Motivación: tipo simple, cola, cola con prioridad, lista
- Abstracción en Programación
- Abstracción Procedimental
  - Especificación
  - Tipos abstracto y representado
- Abstracción por parametrización
- Abstracción de Iteración
- Abstracción de datos

Módulo 3.- TDAs contenedores básicos.

- Pilas
- Colas
- Colas con prioridad
- Conjunto y Bolsa
- Diccionario
- Vectores Dinámicos
- Listas

Módulo 4.- TDAs contenedores complejos.

- Árboles
  - Arboles. Conceptos fundamentales
  - Arboles Generales y Binarios
    - Arboles binarios de búsqueda.
    - # Arboles parcialmente ordenados
- Tablas Hash
- Grafos

Bibliografía

BIBLIOGRAFÍA FUNDAMENTAL:

A. Garrido, J. Fdez-Valdivia, Abstracción y Estructuras de Datos en C++. Delta publicaciones. 2006.

R. Musser, J. Derge y A. Saini. STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library. 3 Edicicion. Adisson-Wesley 2009.

Robson, R. (2013) Using the STL (2nd ed.) Springer Verlag.

BIBLIOGRAFÍA COMPLEMENTARIA:

H.Deitel and P. Deitel. C++ How to program (Early Objects Version) 9/E (2013) Pearson

A. Garrido. Fundamentos de Programación con la STL. Editorial Universidad de Granada, 2015

A. Garrido. Metodología de Programación: de bits a objetos. Editorial Universidad de Granada, 2015

Gilberg, R.F., Forouzan, B.A. (2001). Data structures: A pseudocode approach with C++. Brooks/Cole.

N.M. Josuttis, The C++ Standard Library: A Tutorial and Reference. Addison-Wesley. 1999.