

Departamento de Informática
UNIVERSIDAD DE OVIEDO

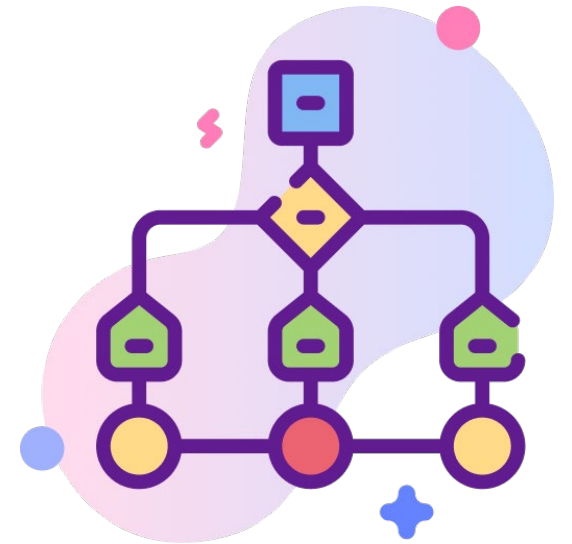
DISEÑO Y ALGORITMIA

Estructura de Datos
2022-2023

María del Rosario Suárez
Fernández

Objetivo

- Resolver problemas de ingeniería
 - Definición del problema
 - Análisis del problema
 - Diseño o desarrollo de un algoritmo
 - Transformación del algoritmo en un programa
 - Identificar los medios para almacenamiento de datos
 - Ejecución y validación del programa
- Medios de almacenamiento → Estructuras de datos



Tipo de dato

- Conjunto de valores que puede asumir una propiedad de una clase
- Los tipos de datos que ya conocemos de un lenguaje de programación:
 - Número entero o real
 - Carácter
 - Booleano
 - Referencia



Estructura de Datos

- Una estructura de datos es una colección de datos que se caracterizan por su organización y las operaciones que se definen en ellos
- Un dato de tipo estructurado es una entidad, con un solo identificador, constituida por datos de otro tipo, de acuerdo con las reglas que definen cada una de las estructuras de datos
- Ejemplos:
 - Cadena → Sucesión de caracteres
 - Matriz → Datos simples organizados en filas y columnas
 - Archivo → Conjunto de registros formados por datos simples

Tipos de Datos Estructurados

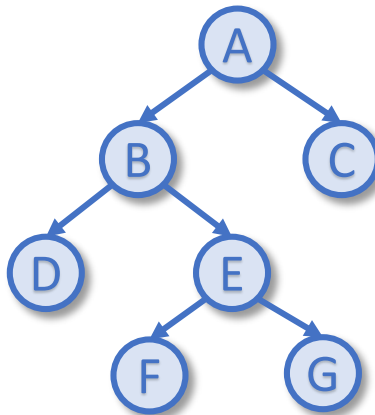
- Según la forma de almacenamiento
 - Estructuras contiguas
 - Cadenas, arrays (vectores y matrices), registros (formados por campos)
 - Estructuras enlazadas
 - Los datos no tiene porque situarse de forma contigua en la memoria
 - Se relacionan mediante punteros
- Según la variabilidad de su tamaño
 - Estructuras estáticas
 - Estructuras dinámicas

Principales familias de estructuras de datos

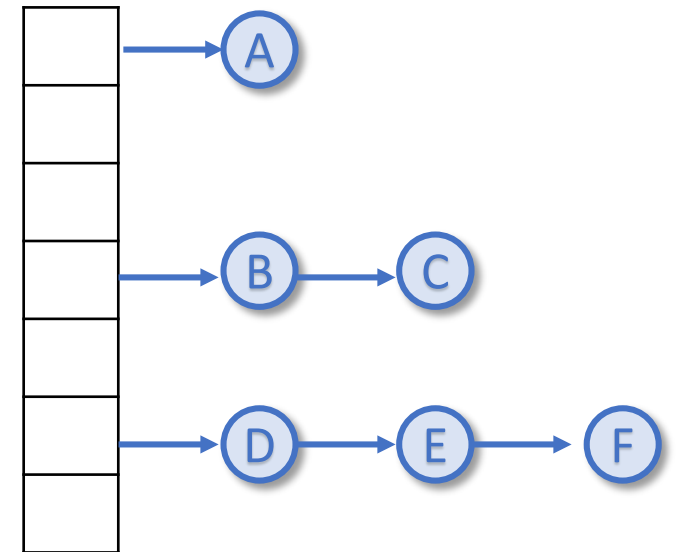
Lineales



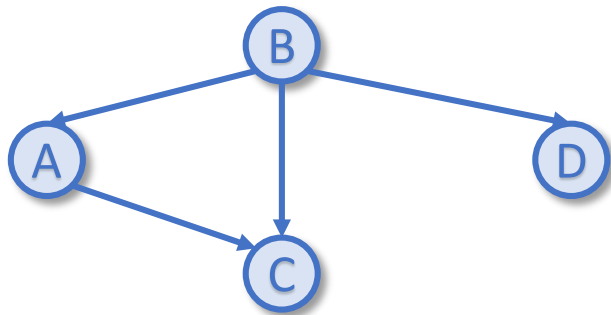
Jerárquicas (Árboles)



Diccionario (tablas hash)



En red (Grafos)



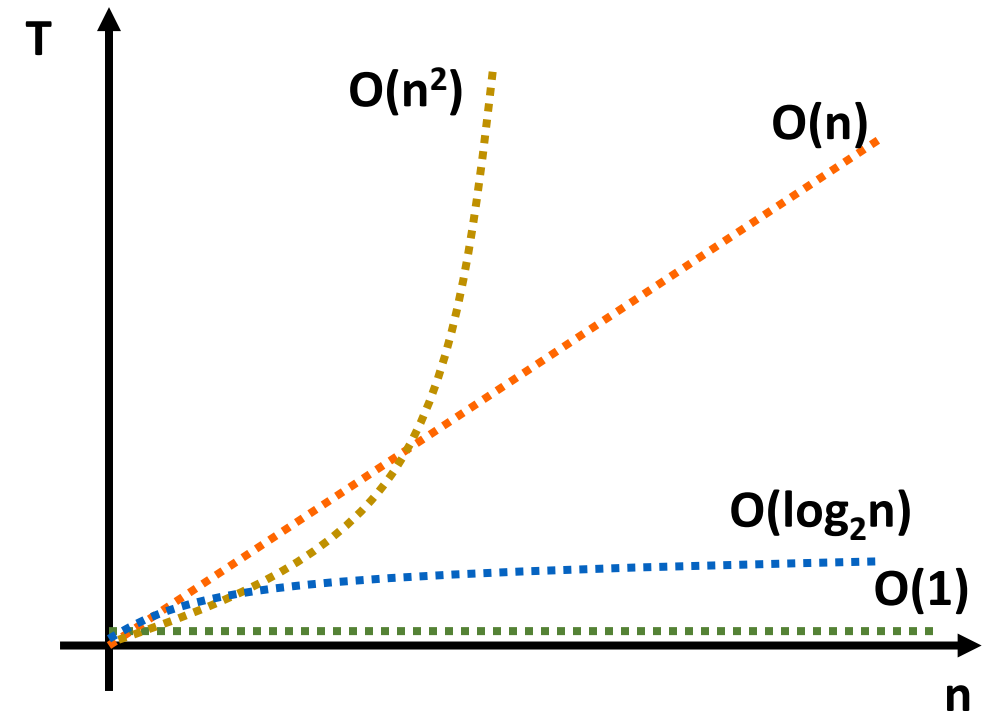
¿Cuál elegir?

Algoritmia y complejidad

- La **algoritmia** consiste en la búsqueda de soluciones a un problema concreto
- Un **algoritmo** es un conjunto ordenado y finito de operaciones simples de tal forma que todas ellas en su conjunto permiten la resolución de un problema
- La **complejidad algorítmica** es una métrica teórica que permite describir el comportamiento de un algoritmo en términos de tiempos de ejecución y de memoria requerida

Cálculo de la complejidad

- Sentencias sencillas $\rightarrow O(1)$
 - Asignaciones, entrada/salida
- Decisión $\rightarrow O(1)$
 - Siempre se toma la peor posible
- Bucles
 - La variable de la iteración aumenta de 1 a $N \rightarrow O(n)$
 - La variable de la iteración aumenta pero reduciendo el número de iteraciones a la mitad en cada caso $\rightarrow O(\log n)$, tomando el entero inmediatamente superior



Orden de complejidad

- $O(1) \rightarrow$ Orden constante
- $O(\log_2 n) \rightarrow$ Orden logarítmico
- $O(n) \rightarrow$ Orden lineal
- $O(n \log_2 n)$
- $O(n^2) \rightarrow$ Orden cuadrático
- $O(n^2 \log_2 n)$
- $O(n^3) \rightarrow$ Orden cúbico
- Problemas de complejidad polinómica (P), polinómica y no determinística (NP) y polinómica, no determinística de extrema complejidad (NP-Complejos)
 - $O(2^n)$ - $O(K^n)$
 - $O(n!)$ y $O(n^n)$

Ejemplos – ¿Cuántas veces se ejecuta....?

```
Algoritmo{  
    test();  
    test();  
  
    int i=3;  
    return (i*test());  
}
```

```
Algoritmo{  
    for (int i=0; i<n; i++) {  
        test();  
        test();  
        test();  
        test();  
        test();  
    }  
  
    test();  
}
```

Ejemplos – ¿Cuántas veces se ejecuta....?

```
Algoritmo{  
  test();  
  test();  
  
  int i=3;  
  return (i*test());  
}
```

3 veces

Complejidad $O(1)$

```
Algoritmo{  
  for (int i=0; i<n; i++) {  
    test();  
    test();  
    test();  
    test();  
    test();  
  }  
  
  test();  
}
```

Ejemplos – ¿Cuántas veces se ejecuta....?

```
Algoritmo{  
  test();  
  test();  
  
  int i=3;  
  return (i*test());  
}
```

3 veces

Complejidad $O(1)$

```
Algoritmo{  
  for (int i=0; i<n; i++) {  
    test();  
    test();  
    test();  
    test();  
    test();  
  }  
  
  test();  
}
```

$5n+1$ veces

Complejidad $O(n)$

Ejemplos – ¿Cuántas veces se ejecuta....?

```
Algoritmo{  
  for (int i=0; i<n; i++)  
    for (int j=0; j<n; j++){  
      test();  
      test();  
    }  
  test();  
}
```

```
Algoritmo{  
  {  
    while (n>0) {  
      test();  
      test();  
      n = n/2;  
    }  
  
    test();  
  }  
}
```

Ejemplos – ¿Cuántas veces se ejecuta....?

```
Algoritmo{  
  for (int i=0; i<n; i++)  
    for (int j=0; j<n; j++){  
      test();  
      test();  
    }  
  test();  
}
```

$2n^2+1$ veces
Complejidad $O(n^2)$

```
Algoritmo{  
  {  
    while (n>0) {  
      test();  
      test();  
      n = n/2;  
    }  
  
    test();  
  }  
}
```

Ejemplos – ¿Cuántas veces se ejecuta....?

```
Algoritmo{  
  for (int i=0; i<n; i++)  
    for (int j=0; j<n; j++){  
      test();  
      test();  
    }  
  test();  
}
```

$2n^2+1$ veces
Complejidad $O(n^2)$

```
Algoritmo{  
  {  
    while (n>0) {  
      test();  
      test();  
      n = n/2;  
    }  
  
    test();  
  }  
}
```

$2(\log_2(n)+1)+1$ veces
Complejidad logarítmica