

Programación Avanzada Unidad I - Introducción

Msc. Víctor Rodríguez Estévez

July 23, 2022



¿Que haremos en este curso?



Nuestro curso

- **Técnicas** avanzadas de programación
- Programación y **resolución de problemas**, con aplicaciones reales.
- **Algoritmos**: método para resolver un problema.
- **Estructura de datos**: método para almacenar información

¿Que haremos en este curso?



Nuestro curso

Tópicos necesarios	Estructura y algoritmo
Tipo de Dato	Pilas, Colas, Listas, Union-find, Colas con prioridad
Técnicas	Divide y vencerás, Programación Dinámica, paradigma funcional.
Ordenamiento	Quick sort, Merge sort, Heap sort
Búsquedas	Árboles binarios, Árboles Rojo-Negro, Tablas Hash
Grafos	Primero en Profundidad, Primero en amplitud, Kruskal, Dijkstra, métodos Heurísticos
Cadenas	Radix sort, tries, expresiones regulares, parsers

¿Por qué estudiar algoritmos?

Su impacto es amplio y de gran alcance

- **Internet:** Búsqueda web...
- **Biología:** Proyecto del genoma humano...
- **Computación:** Automatas, Lenguajes formales, Compiladores
- **Gráficas:** Videojuegos, realidad virtual, ...
- **Seguridad:** Criptografía
- **Interfaz Humano - Computador.** Procesamiento de lenguaje natural, reconocimiento facial, ...
- **Redes sociales:** Recomendaciones, asociaciones, ...
- **Física:** Simulación DFC, Lattice bolzman, N-Body...



¿Por qué estudiar algoritmos?

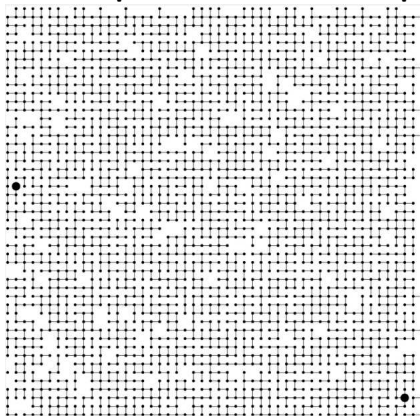


Viejas raíces, nuevas oportunidades

- Algoritmos desde los 300 a. C.
- El estudio de los algoritmos data al menos desde Euclides.
- Formalizado por Church y Turing en la década de 1930.
- Computacionalmente, se plantean infinitos algoritmos importantes.

¿Por qué estudiar algoritmos?

Para resolver problemas que de otro modo no podrían



abordarse.

¿Por qué estudiar algoritmos?



Francis Sullivan

*"Para mí, los grandes algoritmos son la poesía de la computación. Al igual que el verso, pueden ser **concisos, alusivos, densos e incluso misteriosos**. Pero una vez desbloqueados, arrojan una nueva luz brillante sobre algún aspecto de la informática."*



Donald Knuth

"Hay que ver un algoritmo para creerlo. "

¿Por qué estudiar algoritmos?



Linus Torvalds

*"De hecho, afirmaré que la diferencia entre un **mal programador** y uno **bueno**, es si este considera que su código o sus estructuras de datos son más importantes. **Los malos programadores se preocupan por el código.** **Los buenos programadores se preocupan por las estructuras de datos y sus relaciones.**"*



Niklaus Wirth

"Algoritmos + Estructuras de datos = Programas."

¿Por qué estudiar algoritmos?

Pueden descubrir los secretos de la vida y del universo

Los modelos computacionales están reemplazando a los modelos matemáticos en la investigación científica.

Ciencia del siglo XX

- $E = mc^2$
- $F = \frac{Gm_1m_2}{r^2}$
- $[-\frac{\hbar^2}{2m}\nabla^2 + V(r)]\psi(r) = E\psi(r)$

Ciencia del siglo XXI

```
for (double t = 0.0; true; t = t + dt)
  for (int i = 0; i < N; i++){
    bodies[i].resetForce();
    for (int j = 0; j < N; j++)
      if (i != j)
        bodies[i].addForce(bodies[j]);
  }
```

¿Por qué estudiar algoritmos?

Desarrollo profesional



¿Por qué estudiar algoritmos?



Nuestro curso

- Antiguas raíces, nuevas oportunidades.
- Para resolver problemas que de otro modo no podrían resolverse.
- Perfeccionamiento en la resolución de problemas
- Convertirse en un programador competente.
- Descubrir los secretos de la vida y del universo.
- Realización profesional.
- Frustrante a veces, pero altamente aditivo...

Objeto de Estudio



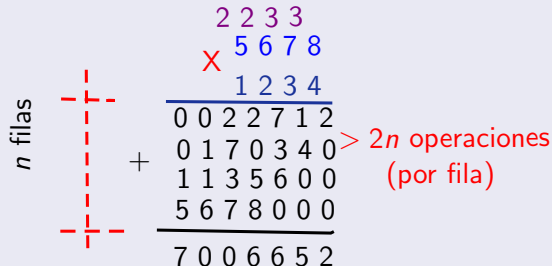
Donde Investigo?

DISEÑO DE ALGORITMOS AVANZADOS
Y
ESTRUCTURAS DE DATOS

Objeto de Estudio

Multiplicando números

- **Entrada:** x, y , dos números de n dígitos.
- **Salida:** $x * y$.



The diagram illustrates the multiplication of two numbers, 2233 and 5678. The multiplier 2233 is written in purple above the multiplicand 5678, which is in blue. A red 'X' is placed to the left of the multiplier. The product is shown in blue below a horizontal line. To the left of the multiplication, a vertical dashed red line is labeled 'n filas' (n rows). To the right of the product, a red arrow points to the text '2n operaciones (por fila)' (2n operations (per row)).

$$\begin{array}{r}
 2233 \\
 \times 5678 \\
 \hline
 0022712 \\
 0170340 \\
 1135600 \\
 5678000 \\
 \hline
 7006652
 \end{array}$$

n filas

$2n$ operaciones (por fila)

Resultado: Numero de operaciones necesarias cn^2 donde c es una constante, y n es el numero de dígitos del multiplicador.



Objeto de Estudio

$\begin{matrix} a & b \\ \boxed{5} & \boxed{6} & \boxed{7} & \boxed{8} \\ \boxed{1} & \boxed{2} & \boxed{3} & \boxed{4} \\ c & d \end{matrix}$

1. ac
2. bd
3. $(a+b)(c+d)$
4. $ad + bc$

$$1. 56 \cdot 12 = 672$$

$$2. 78 \cdot 34 = 2652$$

$$3. (56+78)(12+34) = 6164$$

$$4. 56 \cdot 34 + 12 \cdot 78 = 2840$$

$$\begin{array}{r}
 6720000 \\
 + \quad 2652 \\
 \hline
 2840000 \\
 \hline
 7006652
 \end{array}$$

$$a=5 \quad b=6$$

$$c=1 \quad d=2$$

$$1. 5 \cdot 1$$

$$2. 6 \cdot 2 = 12$$

$$3. (5+6)(1+2) = 33$$

$$4. 5 \cdot 2 + 6 \cdot 1 = 16$$

$$\begin{array}{r}
 500 \\
 + 12 \\
 \hline
 160 \\
 \hline
 672
 \end{array}$$

El Objeto

? Karatsuba

El algoritmo propuesto en la anterior diapositiva, es denominado **algoritmo Karatsuba**. Es una gran ejemplo de la técnica "divide y conquista", por su naturaleza recursiva. Prueba el algoritmo, completa la multiplicación anterior.

Objeto de Estudio



Si podemos hacerlo mejor

- 1 Escribir $x = 10^{\frac{n}{2}}a + b$
- 2 Escribir $y = 10^{\frac{n}{2}}c + d$
- 3 $xy = (10^{\frac{n}{2}}a + b)(10^{\frac{n}{2}}c + d)$
- 4 $xy = (10^{\frac{n}{2}}ac + 10^{\frac{n}{2}}(ad + bc) + bd)$
- 5 **Idea:** Calcular recursivamente ac , ad , bc , bd , luego calcular el caso base.

Objeto de Estudio



Si podemos hacerlo mejor

$$xy = (10^n ac + 10^{\frac{n}{2}}(ad + bc) + bd)$$

- 1 Computar recursivamente ac
- 2 Computar recursivamente bd
- 3 Computar recursivamente $(a + b)(c + d) = ac + bd + ad + bc$
- 4 Truco de Gauss: $(3)-(2)-(1) = ad + ac$

Solo necesita 3 multiplicaciones recursivas (y algunas adiciones)

El Objetivo



Para que Investigo?

APLICA TÉCNICAS AVANZADAS DE DISEÑO DE ALGORITMOS
DE MANERA EFICIENTE, UTILIZANDO ESTRUCTURAS DE DATOS
PARA LA RESOLUCIÓN DE PROBLEMAS

Nuestro curso



Tópicos

- 1 Conviértete en un mejor programador
- 2 Agudiza tus habilidades matemáticas y analíticas
- 3 Empieza a "pensar algorítmicamente"
- 4 Alfabetización en el mundo de las ciencias de la computación
- 5 Haz la diferencia entre un nivel de académico y un aficionado.

El Contenido

Acudimos a la siguiente teoría sobre nuestro objeto de estudio para llegar a nuestro objetivo y así resolver el problema.

UT1: Medición del rendimiento diseño y análisis de algoritmos

- 1 **Medición del rendimiento** diseño y análisis de algoritmos.
- 2 Técnica de diseño de algoritmos: **Dividir y conquistar**.
- 3 Estructuras y Algoritmos para la representación y resolución de **Grafos**.
- 4 Técnica de diseño de algoritmos: **Técnicas Heurísticas**
- 5 Técnica de diseño de algoritmos: **Programación dinámica**
- 6 Algoritmos y Estructuras de Datos: **Geometría Computacional**
- 7 Algoritmos y Estructuras de Datos: **Compiladores**.

El Contenido

Acudimos a la siguiente teoría sobre nuestro objeto de estudio para llegar a nuestro objetivo y así resolver el problema.



UT1: Medición del rendimiento diseño y análisis de algoritmos

- 1 Análisis Asintótico de algoritmos
- 2 Big-O notation.
- 3 Big-Omega
- 4 Big Theta
- 5 Caso de estudio: Union-Find

El Contenido

Acudimos a la siguiente teoría sobre nuestro objeto de estudio para llegar a nuestro objetivo y así resolver el problema.

UT2: Divide y Venceras

- 1 $O(n \log(n))$
- 2 Algoritmos de ordenamiento y búsqueda, estructuras lineales
- 3 Algoritmos de búsqueda, estructuras no lineales
- 4 Algoritmos en Árboles binarios
- 5 Algoritmos en Árboles rojo-negro
- 6 Algoritmos en Árboles balanceados
- 7 Tablas Hash



El Contenido

Acudimos a la siguiente teoría sobre nuestro objeto de estudio para llegar a nuestro objetivo y así resolver el problema.



UT3: Estructuras y Algoritmos para la representación y resolución de **Grafos**

- 1 $O(n \log(n))$
- 2 Grafos unidireccionales
- 3 Grafos dirigidos.
- 4 Camino mas corto
- 5 Algoritmos no Informados.

El Contenido

Acudimos a la siguiente teoría sobre nuestro objeto de estudio para llegar a nuestro objetivo y así resolver el problema.



UT4: Técnicas Heurísticas

- 1 Algoritmos codiciosos
- 2 Heurísticas admisibles.
- 3 Algoritmo A estrella
- 4 Algoritmo de Dijkstra's
- 5 Huffman code
- 6 Caso de estudio: Path Find.

El Contenido

Acudimos a la siguiente teoría sobre nuestro objeto de estudio para llegar a nuestro objetivo y así resolver el problema.



UT5: Programación Dinámica.

- 1 Conjuntos independientes ponderados
- 2 Sub estructura óptima
- 3 Algoritmo de tiempo lineal
- 4 Algoritmo de reconstrucción
- 5 Principios de la programación dinámica.
- 6 Problemas tipo mochila
- 7 Alineación de secuencias

El Contenido

Acudimos a la siguiente teoría sobre nuestro objeto de estudio para llegar a nuestro objetivo y así resolver el problema.



UT6: Geometría Computacional.

- 1 Algoritmo Plane sweep
- 2 Diagramas de Voronoi
- 3 Búsqueda de rango ortogonal
- 4 Aplicaciones.

El Contenido

Acudimos a la siguiente teoría sobre nuestro objeto de estudio para llegar a nuestro objetivo y así resolver el problema.



UT7: Compiladores.

- 1 Técnicas de programación Funcional
- 2 Cadenas y expresiones regulares
- 3 Autómatas finitos
- 4 Análisis sintáctico
- 5 Aplicaciones.

El Contenido



- **Iteración 1:** 18 de Julio al 27 de Agosto:

Unidad Temática	1	2	3	4	5	6	7
UT1	•	•	•	•	•	•	•
UT2	•	•	•	•	•	•	•
UT3	•	•	•	•	•	•	•
Examen	•	•	•	•	•	•	•

El Contenido



- **Iteración 2** : 03 de Septiembre al 22 de Octubre.:

Unidad Temática	1	2	3	4	5	6	7
UT4	•	•	•	•	•	•	•
UT5	•	•	•	•	•	•	•
Examen	•	•	•	•	•	•	•

El Contenido



- **Iteración 3:** 22 de Octubre al 03 de Diciembre.

Unidad Temática	1	2	3	4	5	6	7
UT6	●	●	●	●	●	●	●
UT7	●	●	●	●	●	●	●
Examen	●	●	●	●	●	●	●

La Forma

Espacio y Tiempo donde se realizará el proceso.



Donde y cuando?

- **Tiempo:** 20 semanas (01 de Febrero al 19 de Junio)
- **Aula:** 6 horas académicas a la semana.
- **Virtual:** Lecturas y Diapositivas 6 horas a la semana.
- **Producción:** Resolución de ejercicios : 10 hora semanales.

La Forma

Espacio y Tiempo donde se realizará el proceso.



PYTHON



Java



JavaScript

Donde y cuando?

- **Lenguaje de Programación:** Java | Python
- **IDE:** IntelliJ Idea | PyCharm
- **Testing:** junit | Unit Test
- **Técnica:** TDD
- **Repositorio:** Git
- **Administración de tareas:** Trello.

El Objeto

? Ejemplo:

Es bien conocido que en un acuario algunos peces se pueden comer a otros. Usted tiene un acuario que contiene un cantidad de peces del cual conoce el tamaño. Usted sabe que un pez se puede comer a otro, solo cuando está en el rango de: el doble de tamaño o 10 veces más grande. Se quiere agregar un pez a la pecera, pero queremos determinar el tamaño para no causar conflictos de comerse con otros peces. Considerando esto usted debe escoger un pez que esté entre los siguientes tamaños:

- No hay riesgo de ser comido por otro pez si su tamaño no está entre $\frac{1}{10}$ y $\frac{1}{2}$ inclusive, del tamaño de otro pez.
- No tiene tentación de comerse a otro pez si el tamaño de los otros peces no están entre $\frac{1}{10}$ y $\frac{1}{2}$ inclusive de su tamaño.

Entrada: La entrada consiste de varias líneas. La primera línea de un caso de prueba consiste en el tamaño más pequeño. La segunda línea consiste en el tamaño más grande que puede tener. La tercera línea tiene el número de peces en el acuario. La cuarta línea tiene los tamaños de los peces del acuario separados por un espacio.

Salida: Escriba en una línea el número de tamaños que puede hallar y que no causen conflictos entre peces.

El Objeto



Ejemplo:

Entrada: menor, mayor, peces[]

Salida: aceptados

```

1: aceptados = 0
2: para i desde menor hasta mayor hacer
3:     para p desde 0 hasta peces.largo hacer
4:         p2 = peces[p]
5:         si  $\frac{p2}{2} \geq p \geq \frac{p2}{10}$  or  $\frac{p}{2} \geq p2 \geq \frac{p}{10}$  entonces
6:             salir
7:         fin si
8:     si p == peces.largo-1 entonces
9:         incrementar(aceptados)
10:    fin si
11: fin para
12: fin para
13: devolver aceptados
  
```

El Objeto

? Implementación:

Implementar el algoritmo en el lenguaje de programación de tu preferencia.

El Objeto

? Ejercicios de discusión:

Se le dará una lista de cadenas. Cada carácter de la entrada será calculado como sigue: clave = (Posición en el alfabeto) + (número de elemento) + (posición en el elemento). Todas las posiciones comienzan con 0, por ejemplo la letra A tiene la posición 0, la B la 1, así sucesivamente. La respuesta esperada es la suma de los caracteres en la entrada. Por ejemplo si tenemos CBA, DDD, cada elemento sería calculado como sigue:

- $2 = 2 + 0 + 0$: C en el elemento 0 posición 0
- $2 = 1 + 0 + 1$: B en el elemento 0 posición 1
- $2 = 0 + 0 + 2$: A en el elemento 0 posición 2
- $4 = 3 + 1 + 0$: D en el elemento 1 posición 0
- $5 = 3 + 1 + 1$: D en el elemento 1 posición 1
- $6 = 3 + 1 + 2$: D en el elemento 1 posición 2

El resultado final: $2 + 2 + 2 + 4 + 5 + 6 = 21$.

Entrada: La entrada consiste de varios casos de prueba. Cada caso de prueba viene en una línea que contiene todos los elementos separados por un espacio. La entrada termina cuando no hay más datos.

Salida: Por cada caso de prueba escriba en una línea la clave encontrada con el procedimiento anterior

Casos de prueba: entrada: CBA DDD, salida: 21; entrada: Z, salida: 25; entrada: A B C D E F, salida: 30; entrada: ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ, salida: 1326; entrada: ZZZZZZZZZZ, salida 295; entrada: ES WU DD WC PP OM OU VW DT JL YN VM WI YI OA YE ZV HA RG BO WA YM NJ GJ WA RU, salida 1364.

El Método

Como utilizamos el contenido para llegar al objetivo?



Como?

- Aprendizaje basado en problemas (Casos de estudio).
- Exposición y resolución de ejercicios colaborativos.
- E-learning.
- Coding Dojo (Kata - Randori).
- Modelar, resolver, programar y experimentar!!!!

La Evaluación

En que medida nos acercamos al objetivo?



Evaluación

- Pruebas Diagnosticas.
- **Resoluciones en Red:** Resoluciones colaborativas, participación, Foros, etc.
- **Evaluación Continua** - Desarrollo de Prácticas Semanales y Aplicación.
- **Tres iteraciones:** Tres Pruebas Objetivas.

La Evaluación

En que medida nos acercamos al objetivo?



Evaluación

- **Primera Iteración:** 18 de Marzo.
- **Segunda Iteración :** 6 de Mayo.
- **Tercera Iteración:** 17 de Junio.

Bibliografía

Bibliografía

- Algorithms, S. Dasgupta, C. H. Papadimitriou, and U. V. Vaziran.
- The Design and Analysis of Computer Algorithm, Alfred V.Aho, Jhon Hopcroft, Jeffrey Ullman.
- Data Structures and Algorithms, Alfred V.Aho, Jhon Hopcroft, Jeffrey Ullman.

Conclusión

Conclusiones

- Discutimos la **importancia de los algoritmos**
- Revisamos el **enfoque de algoritmos** que tienen algunos estudiosos del mismo.
- Conversamos acerca de la relación de las **estructuras de datos y los algoritmos**.
- Identificamos que pueden existir varias soluciones, pero **unas mas eficientes que otras**.
- Establecimos nuestro **objeto de estudio**.
- Establecimos la **competencia del curso**.
- Recordamos **elementos de programación y java**.