Algoritmos
Estructuras de Datos I

Facultad de Ciencias Exactas y Tecnología

Universidad Nacional de Tucumán

2025

Unidad I

Algoritmos.

Definición. Características. Complejidad de algoritmos.

Notación O grande. Técnicas de análisis de algoritmos.

Técnicas para el diseño de algoritmos: recursión, divide & conquer, balance, programación dinámica, técnica greedy.

Rendimiento de un programa. Complejidad de espacio. Complejidad de tiempo.

Complejidad de un problema. Introducción a Clases de problemas P y NP, y problemas NP completos.

Unidad I

Muchas personas que no estan familiarizadas con estudios matemáticos, se imaginan que puesto que su meta (la máquina analítica de Babbage) es dar sus resultados en notación numérica, la naturaleza de sus procedimientos deben ser por consiguiente aritméticos y numéricos, más que algebráicos y analíticos. Esto es un error. La máquina puede disponer y combinar sus cantidades numéricas tal como si fuesen letras u otros símbolos generales cualesquiera; y de hecho podría proporcionar sus resultados en notación algebráica, si se hicieran las previsiones convenientes.

—ADA AUGUSTA, Condesa de Lovelace (1844)

Del libro: "*El arte de programar ordenadores, Algoritmos Fundamentales*" , (The Art of Computer Programming)

Donald Knuth. Editorial Reverté. (1985)

Unidad I

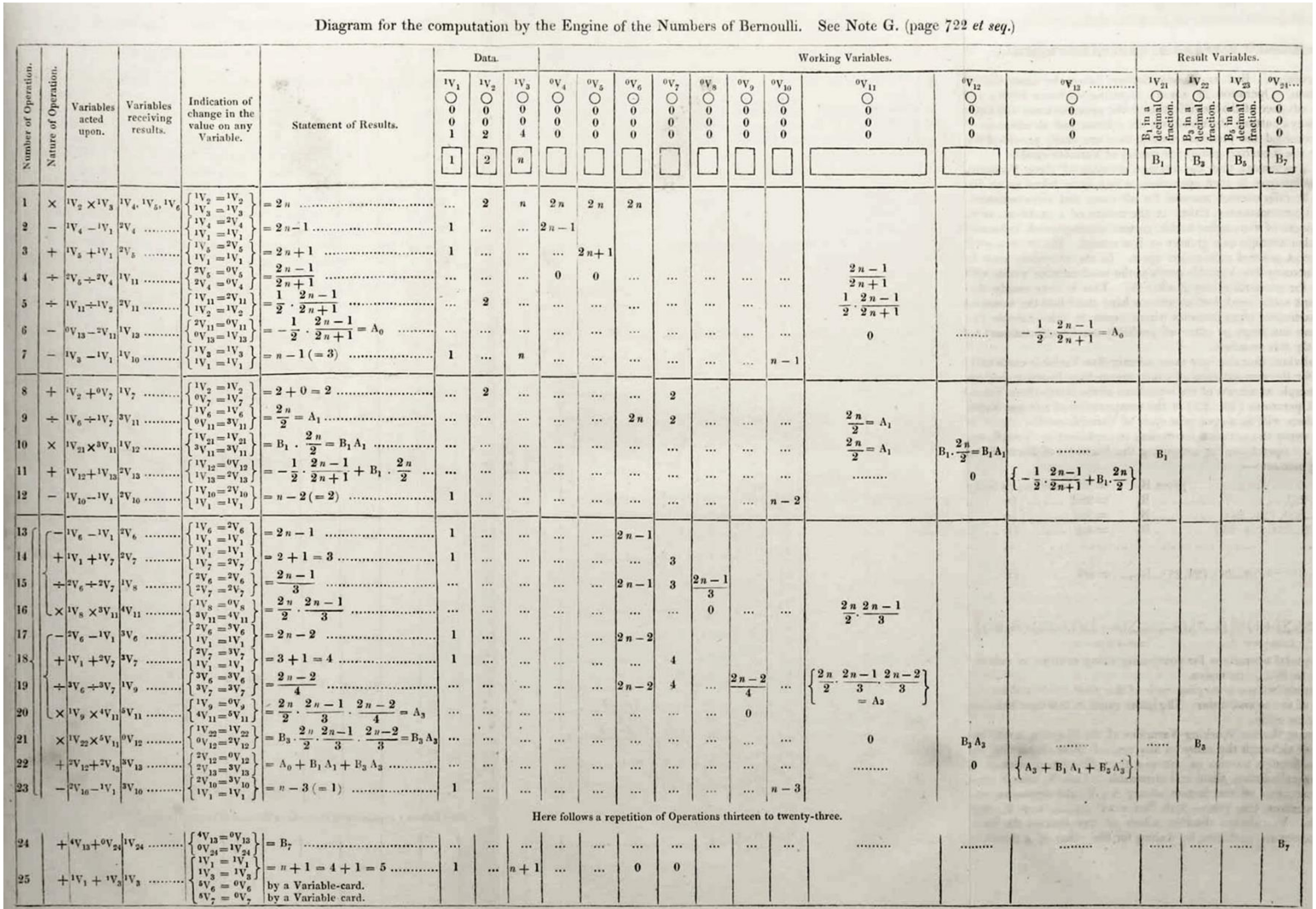


Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

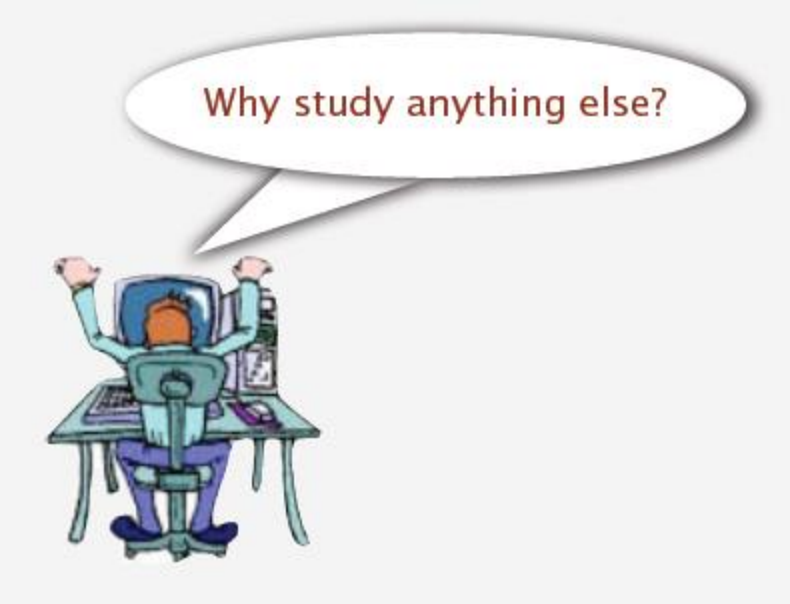
Number of Operation.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data. $^1V_1$ 0 0 0 1 $\boxed{1}$	$^1V_2$ 0 0 0 2 $\boxed{2}$	$^1V_3$ 0 0 0 4 $\boxed{n}$	Working Variables. $^0V_4$ 0 0 0 0	$^0V_5$ 0 0 0 0	$^0V_6$ 0 0 0 0	$^0V_7$ 0 0 0 0	$^0V_8$ 0 0 0 0	$^0V_9$ 0 0 0 0	$^0V_{10}$ 0 0 0 0	$^0V_{11}$ 0 0 0 0	$^0V_{12}$ 0 0 0 0	$^0V_{13}$ 0 0 0 0	Result Variables. $^1V_{21}$ B_1 in a decimal fraction. $\boxed{B_1}$	$^1V_{22}$ B_3 in a decimal fraction. $\boxed{B_3}$	$^1V_{23}$ B_5 in a decimal fraction. $\boxed{B_5}$	$^0V_{24}$ 0 0 0 0 $\boxed{B_7}$
1	×	$^1V_2 \times {}^1V_3$	$^1V_4, {}^1V_5, {}^1V_6$	$^1V_2 = {}^1V_2$, $^1V_3 = {}^1V_3$	$= 2n$	...	2	n	$2n$	$2n$	$2n$											
2	−	$^1V_4 - {}^1V_1$	$^2V_4$	$^1V_4 = {}^2V_4$, $^1V_1 = {}^1V_1$	$= 2n - 1$	1	...	...	$2n-1$													
3	+	$^1V_5 + {}^1V_1$	$^2V_5$	$^1V_5 = {}^2V_5$, $^1V_1 = {}^1V_1$	$= 2n + 1$	1	...	...	...	$2n+1$												
4	÷	$^2V_5 \div {}^2V_4$	$^1V_{11}$	$^2V_5 = {}^0V_5$, $^2V_4 = {}^0V_4$	$= \frac{2n-1}{2n+1}$	...	...	...	0	0	...	...	...	...	...	$\frac{2n-1}{2n+1}$						
5	÷	$^1V_{11} \div {}^1V_2$	$^2V_{11}$	$^1V_{11} = {}^2V_{11}$, $^1V_2 = {}^1V_2$	$= \frac{1}{2} \cdot \frac{2n-1}{2n+1}$	...	2	...	...	...	...	...	...	...	...	$\frac{1}{2} \cdot \frac{2n-1}{2n+1}$						
6	−	$^0V_{13} - {}^2V_{11}$	$^1V_{13}$	$^2V_{11} = {}^0V_{11}$, $^0V_{13} = {}^1V_{13}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$	...	...	...	...	...	...	...	...	...	...	0	....	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$				
7	−	$^1V_3 - {}^1V_1$	$^1V_{10}$	$^1V_3 = {}^1V_3$, $^1V_1 = {}^1V_1$	$= n - 1 \ (= 3)$	1	...	n	...	...	...	...	...	...	$n-1$							
8	+	$^1V_2 + {}^0V_7$	$^1V_7$	$^1V_2 = {}^1V_2$, $^0V_7 = {}^1V_7$	$= 2 + 0 = 2$	...	2	...	...	...	...	2										
9	÷	$^1V_6 \div {}^1V_7$	$^3V_{11}$	$^1V_6 = {}^1V_6$, $^0V_{11} = {}^3V_{11}$	$= \frac{2n}{2} = A_1$	...	...	...	...	...	$2n$	2	...	...	...	$\frac{2n}{2} = A_1$						
10	×	$^1V_{21} \times {}^3V_{11}$	$^1V_{12}$	$^1V_{21} = {}^1V_{21}$, $^3V_{11} = {}^3V_{11}$	$= B_1 \cdot \frac{2n}{2} = B_1 A_1$	...	...	...	...	...	...	...	...	...	...	$\frac{2n}{2} = A_1$	$B_1 \cdot \frac{2n}{2} = B_1 A_1$	....	B_1			
11	+	$^1V_{12} + {}^1V_{13}$	$^2V_{13}$	$^1V_{12} = {}^0V_{12}$, $^1V_{13} = {}^2V_{13}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}$	...	...	...	...	...	...	...	...	...	...	....	0	$\left\{-\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}\right\}$				
12	−	$^1V_{10} - {}^1V_1$	$^2V_{10}$	$^1V_{10} = {}^2V_{10}$, $^1V_1 = {}^1V_1$	$= n - 2 \ (= 2)$	1	...	...	...	...	...	...	...	...	$n-2$							
13	−	$^1V_6 - {}^1V_1$	$^2V_6$	$^1V_6 = {}^2V_6$, $^1V_1 = {}^1V_1$	$= 2n - 1$	1	...	...	...	...	$2n-1$											
14	+	$^1V_1 + {}^1V_7$	$^2V_7$	$^1V_1 = {}^1V_1$, $^1V_7 = {}^2V_7$	$= 2 + 1 = 3$	1	...	...	...	...	...	3										
15	÷	$^2V_6 \div {}^2V_7$	$^1V_8$	$^2V_6 = {}^2V_6$, $^2V_7 = {}^2V_7$	$= \frac{2n-1}{3}$	...	...	...	...	...	$2n-1$	3	$\frac{2n-1}{3}$									
16	×	$^1V_8 \times {}^3V_{11}$	$^4V_{11}$	$^1V_8 = {}^0V_8$, $^3V_{11} = {}^4V_{11}$	$= \frac{2n}{2} \cdot \frac{2n-1}{3}$	...	...	...	...	...	...	...	0	...	...	$\frac{2n}{2} \cdot \frac{2n-1}{3}$						
17	−	$^2V_6 - {}^1V_1$	$^3V_6$	$^2V_6 = {}^3V_6$, $^1V_1 = {}^1V_1$	$= 2n - 2$	1	...	...	...	...	$2n-2$											
18	+	$^1V_1 + {}^2V_7$	$^3V_7$	$^2V_7 = {}^3V_7$, $^1V_1 = {}^1V_1$	$= 3 + 1 = 4$	1	...	...	...	...	...	4										
19	÷	$^3V_6 \div {}^3V_7$	$^1V_9$	$^3V_6 = {}^3V_6$, $^3V_7 = {}^3V_7$	$= \frac{2n-2}{4}$	...	...	...	...	...	$2n-2$	4	...	$\frac{2n-2}{4}$	...	$\left\{\frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{3} = A_3\right\}$						
20	×	$^1V_9 \times {}^4V_{11}$	$^5V_{11}$	$^1V_9 = {}^0V_9$, $^4V_{11} = {}^5V_{11}$	$= \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4} = A_3$	...	...	...	...	...	...	...	...	0								
21	×	$^1V_{22} \times {}^5V_{11}$	$^0V_{12}$	$^1V_{22} = {}^1V_{22}$, $^0V_{12} = {}^2V_{12}$	$= B_3 \cdot \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{3} = B_3 A_3$	...	...	...	...	...	...	...	...	...	...	0	$B_3 A_3$	....	......	B_3		
22	+	$^2V_{12} + {}^2V_{13}$	$^3V_{13}$	$^2V_{12} = {}^0V_{12}$, $^2V_{13} = {}^3V_{13}$	$= A_0 + B_1 A_1 + B_3 A_3$	...	...	...	...	...	...	...	...	...	...	....	0	$\left\{A_3 + B_1 A_1 + B_3 A_3\right\}$				
23	−	$^2V_{10} - {}^1V_1$	$^3V_{10}$	$^2V_{10} = {}^3V_{10}$, $^1V_1 = {}^1V_1$	$= n - 3 \ (= 1)$	1	...	...	...	...	...	...	...	...	$n-3$							
					Here follows a repetition of Operations thirteen to twenty-three.																	
24	+	$^4V_{13} + {}^0V_{24}$	$^1V_{24}$	$^4V_{13} = {}^0V_{13}$, $^0V_{24} = {}^1V_{24}$	$= B_7$	...	...	...	...	...	...	...	...	...	...	....	....	....	......	......	......	B_7
25	+	$^1V_1 + {}^1V_3$	$^1V_3$	$^1V_1 = {}^1V_1$, $^1V_3 = {}^1V_3$, $^5V_6 = {}^0V_6$, $^5V_7 = {}^0V_7$	$= n + 1 = 4 + 1 = 5$ by a Variable-card. by a Variable card.	1	...	$n+1$	...	...	0	0										

Primer algoritmo publicado, cálculo de los números de Bernoulli, Charles Babbage y Ada Augusta Lovelace,1843.

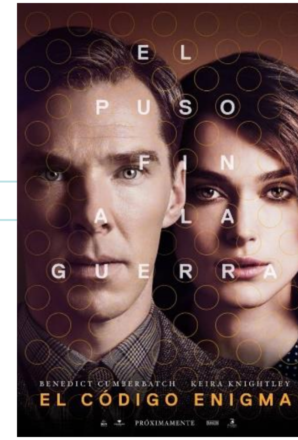
Algoritmos

Por qué estudiar hoy algoritmos?

- Por su amplio impacto y posibilidades actuales y futuras: internet, computadoras, gráficos y videos, seguridad, multimedia, redes sociales, biología, física, etc. etc.
- Sus antiguas raíces y nuevas oportunidades de aplicación.
- Para resolver problemas que no podrían resolverse de otra manera.
- Para ser un programador competente.
- Para entretenerse.
- Para obtener ganancias.
- …



Algoritmos



Por qué estudiar algoritmos?

AYER

Alan Turing (1912-1954), matemático dedicado a la computación.

- Es considerado uno de los padres de la informática.
- Aportó a la formalización de los conceptos de algoritmos y computación.
- Durante la 2da guerra mundial , usó las computadoras para descifrar el lenguaje secreto de los códigos de comunicación utilizado por los nazis con la máquina Enigma. Con esto contribuyó según historiadores a nada menos que acortar la Segunda Guerra Mundial.

Algoritmos

Por qué estudiar algoritmos?

- Para salvar a la humanidad desarrollando algoritmos y programando simulaciones en la supercomputadora más poderosa del mundo colaborando en la búsqueda de una cura para el Coronavirus.
- La supercomputadora Summit de IBM, la más poderosa e inteligente del mundo, desde Marzo 2020 participó en la búsqueda de una vacuna contra el coronavirus. Los investigadores del Laboratorio Nacional Oak Ridge (ORNL) del Departamento de Energía de EE.UU. Utilizaron esta máquina para realizar simulaciones a una velocidad sin precedentes, un trabajo que ya está dando sus primeros frutos. En solo dos días, Summit identificó y estudió 77 compuestos potenciales de fármacos para luchar contra el Covid-19. Esta tarea habría llevado años en un laboratorio tradicional.
- Según explican desde IBM, este es un proceso lento sin computadoras potentes que puedan realizar simulaciones digitales para reducir el rango de variables potenciales.
- https://www.clarin.com/viste/coronavirus-supercomputadora-poderosa-mundo-acelera-busqueda-vacuna_0_40t7DMs5k.html

Algoritmos

Para obtener ganancias …

Significado de Algoritmo

- **El Diccionario de la Real Academia Española (RAE):**

- ***algoritmo.*** (Quizá del lat. tardío **algobarismus,* y este abrev. del ár. clás. *ḥisābu lḡubār*, cálculo mediante cifras arábigas). m. Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. || 2. Método y notación en las distintas formas del cálculo.

- ***algorítmico*, ca.** adj. Perteneciente o relativo al algoritmo.

- ***algoritmia.*** *(De algoritmo). f. Ciencia que estudia los algoritmos.*

Significado de Algoritmo

Concepto moderno de Algoritmo:

- Proceso
- Método
- Receta
- Técnica
- Procedimiento
- Rutina
- Método de cálculo
- …

En Informática el concepto de algoritmo está asociado a algo mas riguroso que a una secuencia de operaciones.

Definición de Algoritmo

Un ***algoritmo*** *es una especificación rigurosa* ***de la secuencia finita de instrucciones*** *a ejecutar sobre un autómata para* ***resolver un problema en un tiempo finito,*** *con las siguientes condiciones:*

- **Entrada:** puede ser vacía.
- **Salida:** no puede ser vacía.
- **Definido:** cada instrucción debe definirse de modo preciso, debe ser clara, rigurosa, no ambigua.
- **Efectivo:** cada instrucción debe ser factible, deben poder ejecutarse de modo exacto y en un lapso finito de tiempo.
- **Finito:** el algoritmo debe terminar en un numero finito de pasos.
- **Determinístico:** siempre la misma salida para idénticas entradas.

Algoritmo de Euclides

Definición de máximo común divisor (MCD):

Sean a, b $\in$ Z. El máximo común divisor entre a y b, que se nota (a : b) o mcd(a, b), es el mayor de los divisores comunes de a y b.

El Algoritmo de Euclides es uno de los métodos más antiguos en la historia de la matemática, creado alrededor del año 300 AC, por el matemático griego Euclides (325-265 AC).

Su principal objetivo es encontrar el máximo común divisor (MCD) de dos números enteros.

Algoritmo de Euclides

Entrada: m,n, dos numeros enteros positivos

Salida: m, entero positivo MCD de m y n
Auxiliar: r, entero positivo o nulo

E1. Leer (m,n)
E2. Mientras n ≠ 0 Hacer
    r ← resto (m/n)
    m ← n
    n ← r
E3. Escribir (m)
E4. Fin

Es un algoritmo ?

Algoritmo de Euclides

- Para responder a la pregunta si es un algortimo, considere el siguiente teorema.

Teorema:

El algoritmo de Euclides siempre termina y el último resto no nulo que se obtiene es el máximo común divisor entre a y b.

Algoritmo Criba de Eratostenes

Un número primo es un número entero que tiene exactamente dos divisores: el 1 y el mismo número.

La criba de Eratóstenes (matemático griego, siglo II AC) es un algoritmo que permite hallar todos los números primos menores que un número natural n dado.

Se forma una lista con todos los números naturales comprendidos entre 2 y $\sqrt{n}$, y se van tachando los números que no son primos de la siguiente manera: Comenzando por el 2, se tachan todos sus múltiplos; comenzando de nuevo con el siguiente, cuando se encuentra un número entero que no ha sido tachado, ese número es declarado primo, y se procede a tachar todos sus múltiplos, así sucesivamente.

El proceso termina cuando el cuadrado del siguiente número confirmado como primo es mayor que n

Algoritmo Criba de Eratostenes

Entrada: n, numero entero positivo

Salida: lista de numeros primos menores o iguales a n

C1. Leer (n)
C2. Escribir lista de números naturales de 1 a n
C3. Para i desde 2 hasta $\lfloor \sqrt{n} \rfloor$ hacer
    Si i no ha sido tachado entonces
        Para j desde i hasta n/i hacer
            Tachar el numero $i*j$
C4. Escribir (lista de números no tachados)
C5. Fin

Es un algoritmo ?

Algoritmo Criba de Eratostenes

Ejemplo: n=100

C1:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Algoritmo Criba de Eratostenes

Ejemplo: n=100

C3. Se tachan los múltiplos de 2

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Algoritmo Criba de Eratostenes

Ejemplo: n=100

C3. Se tachan los múltiplos de 3

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Algoritmo Criba de Eratostenes

Ejemplo: n=100

C3. Se tachan los múltiplos de 5

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Algoritmo Criba de Eratostenes

Ejemplo: n=100

C3. Se tachan los múltiplos de 7

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Algoritmo Criba de Eratostenes

Resultado:

Los primos menores que 100 son 25 numeros:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Características de un Algoritmo

*Un **algoritmo** debe ser:*

- **Correcto**
- **Simple**
- **Elegante**
- **Eficiente**
- **Adaptable**

Problema de Multiplicación

Problema:
Multiplicar dos enteros positivos a y b.

Este problema se puede resolver con varios algoritmos:

- Multiplicación clásica
- Multiplicación sumas sucesivas
- Multiplicación en bloque
- Multiplicación hindú o Fibonacci
- Multiplicación a la rusa
- Multiplicación egipcia
- Multiplicación D&C
- …

Cual algoritmo es "***mejor***" ?

Multiplicación clásica

Ej. a=981 , b=12

Para calcular 981x12 se multiplica cifra por cifra y suma:

```
      981
  x
       12
 ---------
   1962
+  981
 ---------
  11772
```

a tiene m cifras

b tiene n cifras

total de operaciones = m x n multiplicaciones + 1 suma

Multiplicación sumas sucesivas

1) Sumar b veces el numero a :

$$a \times b = a + a + \ldots. + a \quad \text{(b veces)}$$

total de operaciones = b sumas

Ej. a=981, b=12

981 x 12 = 981+981+…+981=**11772**

2) Sumar a veces el numero b:

$$a \times b = b + b + b + \ldots + b \quad \text{(a veces)}$$

total de operaciones = a sumas

Ej. a=981, b=12

981 x 12 = 12+12+…+12=**11772**

Multiplicación en bloque

Ej. **981 x 12**

Descomponer ambos números en base 10:

981= 900+80+1

12 = 10+2

Multiplicar en tabla:

	900	80	1
10	9000	800	10
2	1800	160	2

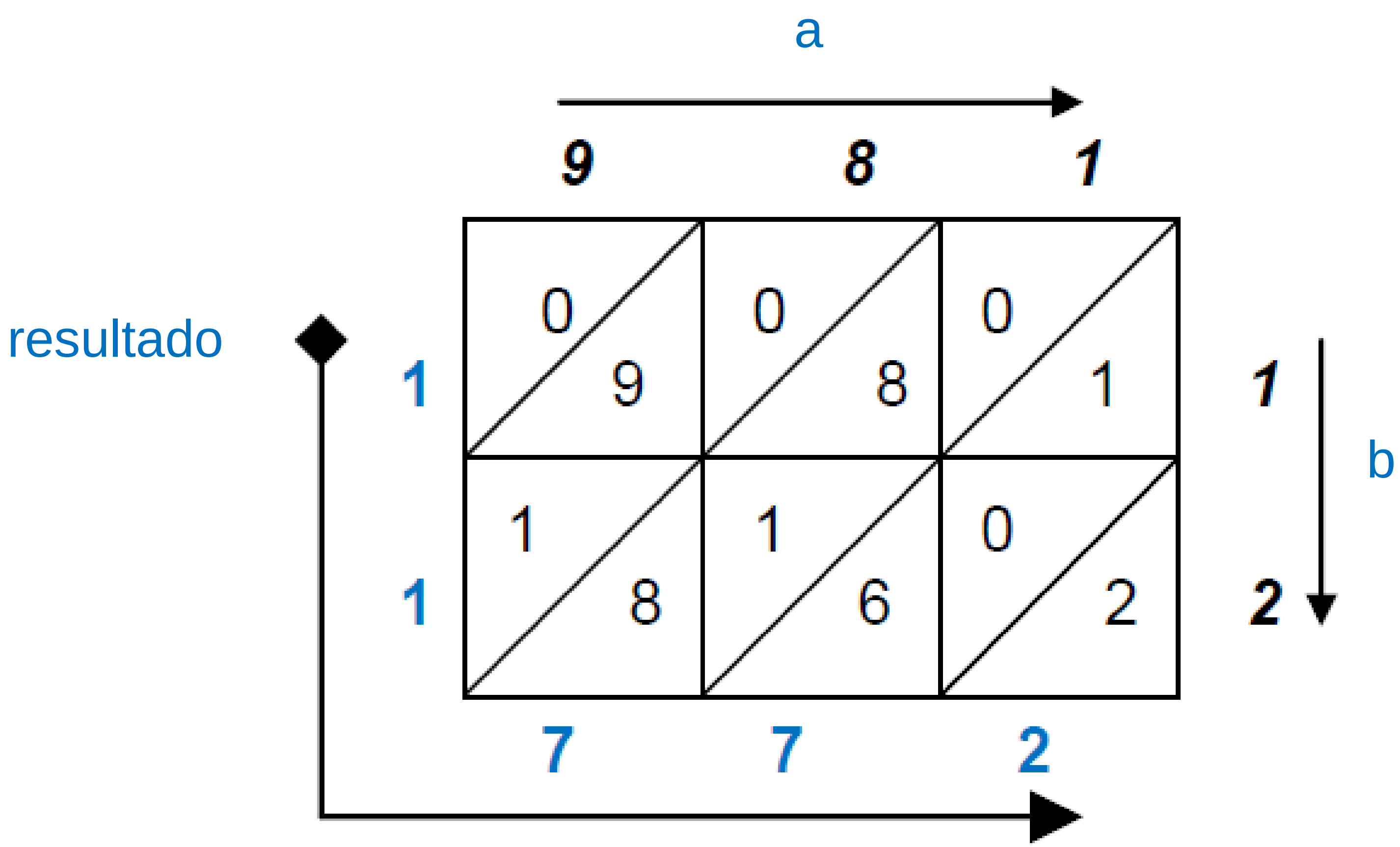
Sumar:

9000+1800+800+160+10+2=**11772**

Multiplicación hindú o de Fibonacci

Multiplica cifra por cifra en una tabla y suma:

Ej. 981 x 12



Multiplicación a la rusa

Se arma una tabla de 3 columnas:

- Comenzando con el numero **a** se lo divide en 2 hasta llegar a 1.
- Comenzando con el numero **b** se lo multiplica por 2.
- Si **a** es impar se suma el correspondiente valor de **b**

Ej. 981 x 12

a	b	si a es impar suma b
981	12	12
a/2	b*2	
490	24	
245	48	48
122	96	
61	192	192
30	384	
15	768	768
7	1536	1536
3	3072	3072
1	6144	6144
		total=11772

Multiplicación egipcia

Se pasa el numero **a** a binario y se escribe su desarrollo en potencias de 2.
Ej. 981 x 12

a = $(981)_{10}$ = $(1111010101)_2$

= $1.2^0+0.2^1+1.2^2+0.2^3+1.2^4+0.2^5+1.2^6+1.2^7+1.2^8+1.2^9$

n	2^n	coef(a)	b*2	Si coef(a) es 1 suma b
0	1	1	12	12
1	2	0	24	
2	4	1	48	48
3	8	0	96	
4	16	1	192	192
5	32	0	384	
6	64	1	768	768
7	128	1	1536	1536
8	256	1	3072	3072
9	512	1	6144	6144
				total=11772

Complejidad de Multiplicar

- Los algoritmos realizados manualmente tienen una complejidad asintótica de $O(n^2)$
- En 1960 Anatoly Karatsuba descubrió que era posible lograr una menor complejidad con su algoritmo.
- En 1968, se publica el algoritmo de Schönhage-Strassen, que hace uso de la tranformada de Fourier, consiguiendo $O(n \log n \log \log n)$
- En 2007 Martin Fürer, mejoró la complejidad asintótica de la multiplicación de números enteros utilizando transformadas de Fourier sobre números complejos.
- En 2014, Harvey, Joris van der Hoeven y Lecerf presentaron un nuevo algoritmo que logra un mejor tiempo de ejecución . También propusieron una variante de su algoritmo un poco mas eficiente cuya validez se basa en conjeturas estándar sobre la distribución de los primos de Mersenne .

Complejidad de Multiplicar

- En 2016, Covanov y Thomé propusieron un algoritmo de multiplicación de enteros basado en una generalización de los primos de Fermat que, alcanza el mismo limite de complejidad. Esto coincide con el resultado de 2015 de Harvey, van der Hoeven y Lecerf, pero utiliza un algoritmo diferente y se basa en una conjetura diferente.

- En 2019, David Harvey y Joris van der Hoeven anunciaron el descubrimiento de un algoritmo de multiplicación de **O (n log n).** Se publicó en Annals of Mathematics en 2021.-