

Principios y Herramientas de Programación

Dra. Jessica Andrea Carballido

jac@cs.uns.edu.ar

```
opcion;
printf("1. Capital de Argentina\n");
printf("2. Capital de España\n");
printf("3. 10000+58000 = ?\n");
printf("4. Capital de Uruguay\n");
scanf("%i",&opcion);
switch(opcion)
{
case 1:
printf("\n\nBuenos Aires");
break;
case 2:
printf("\n\nMadrid");
break;
case 3:
printf("\n\n68000");
break;
case 4:
printf("\n\nMontevideo");
break;
default:
printf("\n\nOpcion erronea. Intente
```

Dpto. de Ciencias e Ingeniería de la Computación

UNIVERSIDAD NACIONAL DEL SUR

Problemas:

- ✓ Sumar los dígitos de un número N .
- ✓ Obtener el promedio de los dígitos de N .
- ✓ Sumar los dígitos de un número N que sean múltiplos de k .
- ✓ Sumar todos los números en el intervalo (a, b) que sean impares.
- ✓ Obtener el promedio de los números contenidos en el intervalo $(a, b]$.
- ✓ Obtener el promedio de los números contenidos en el intervalo $(a, b]$ que sean impares.
- ✓ Contar en el intervalo $(a, b]$ cuántos números son múltiplos del promedio de sus dígitos.
- ✓ Determinar si en el intervalo $(a, b]$ hay algún múltiplo del promedio de sus dígitos.



Dra. Jessica Andrea Carballido

CONICET - DCIC (UNS)



Problema primitiva

Contar la cantidad de números en el intervalo $[a, b]$ cuya suma de dígitos es un múltiplo de 3.

Problema: recorrer todos los números en un intervalo.

Sub-problema: sumar los dígitos de un número.

Recorrer un intervalo y contar elementos en el intervalo que cumplen con una condición

Recorrer un número y sumar sus dígitos



Problema primitiva

Algoritmos

Algoritmo SumaDigitos

DE: N

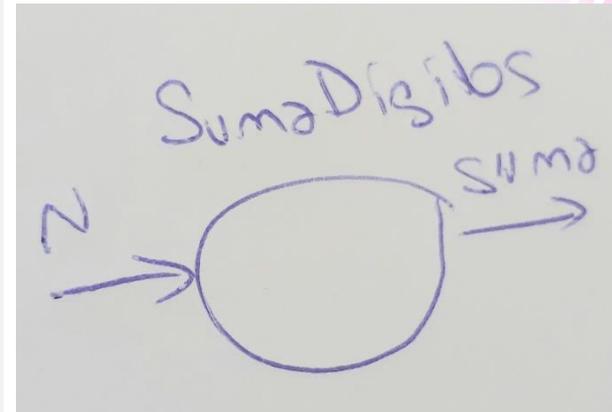
DS: Suma

suma \leftarrow 0

mientras (N > 0) hacer

 suma \leftarrow suma + N mod 10

 N \leftarrow N div 10



Algoritmo CuantosEnSecuencia

DE: a, b

DS: cantidad

cantidad \leftarrow 0

para i desde a hasta b hacer

 Si (**SumaDigitos**(i) mod 3 = 0)

 entonces

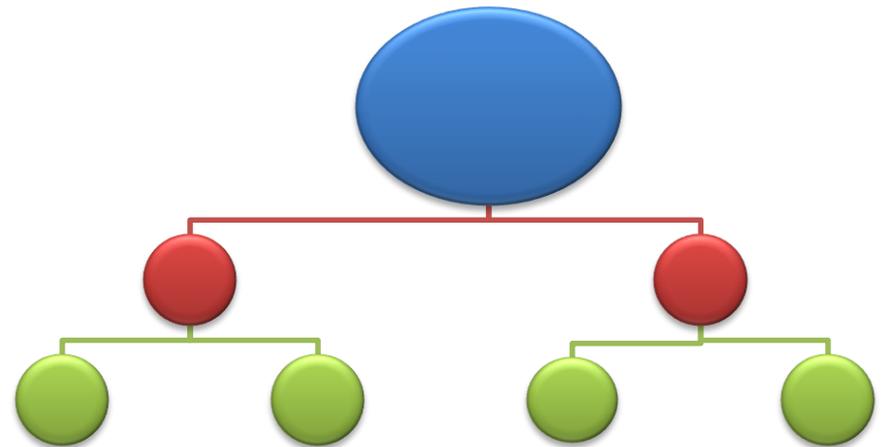
 cantidad \leftarrow cantidad + 1

El valor de *i* se copia como dato de entrada en *N* para la ejecución primitiva en cada invocación.

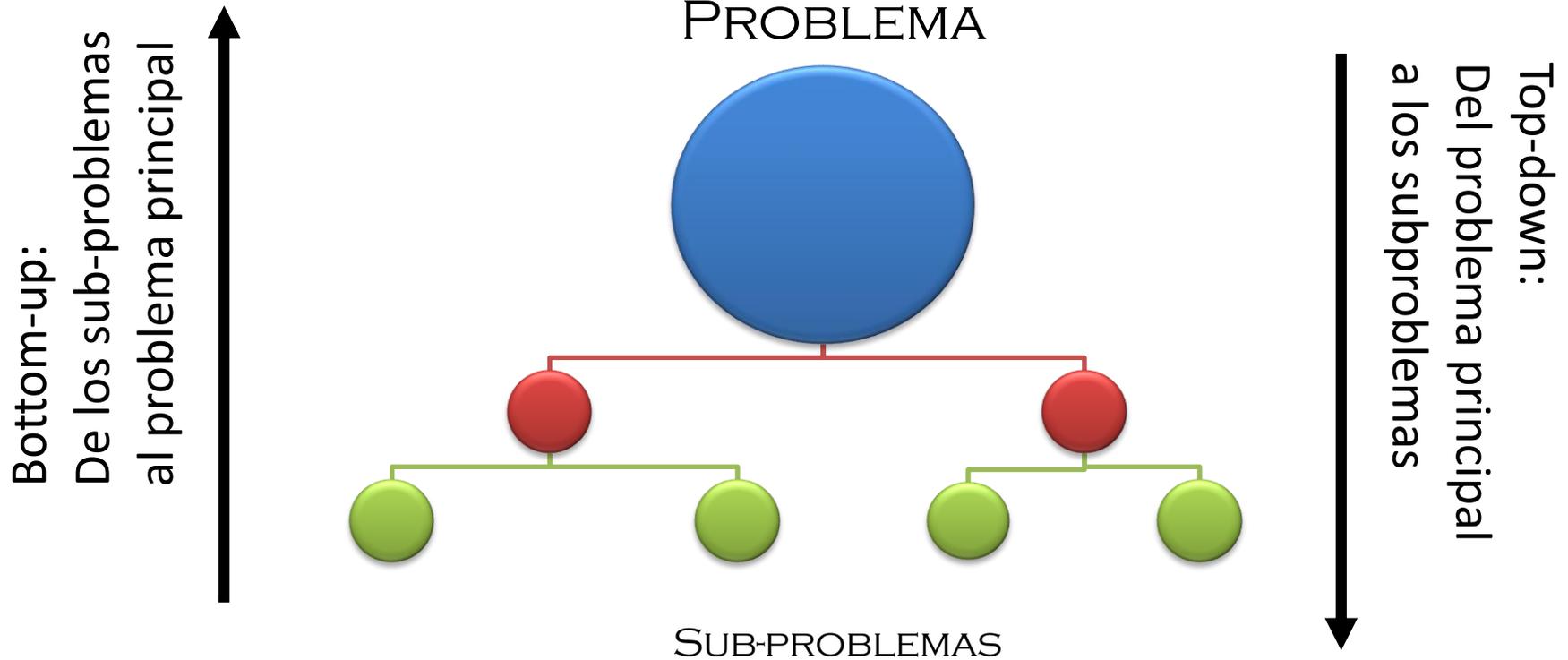
Descomposición de Problemas

Cuando la complejidad de los problemas aumenta, la tarea de hallar una solución se torna más difícil.

Una metodología para reducir la complejidad consiste en plantear la solución del problema a partir de la solución de una serie de sub-problemas más sencillos que forman parte del problema original.

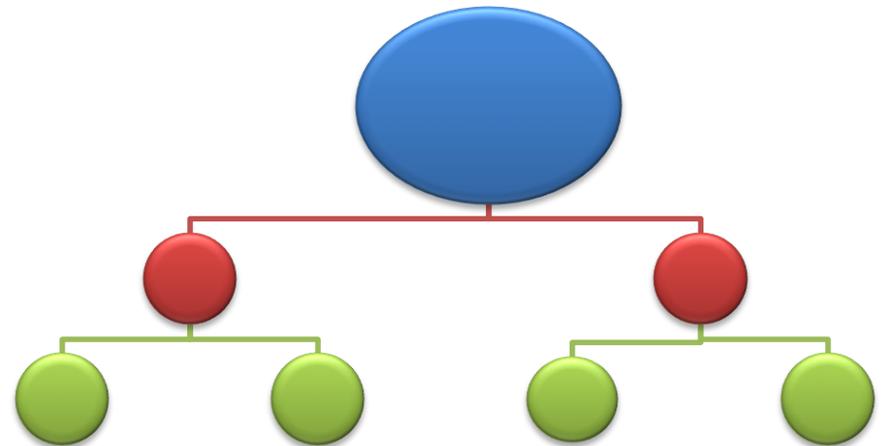


“Bottom-up vs. Top-down”



“Bottom-up vs. Top-down”

- ❏ La técnica *bottom-up* consiste en ir resolviendo en primer lugar los problemas más elementales, usando esas soluciones para resolver problemas mayores.
- ❏ La técnica *top-down* resuelve primeramente el problema mayor, dejando pendiente la solución de los problemas más elementales para más adelante.



Problema

Sucesiones

Escriba un algoritmo para calcular el i -ésimo término de la siguiente sucesión infinita:

$$S = 2/1, 4/2, 8/6, 16/24, 32/120, 64/720\dots$$

Término general: $2^N/N!$



¿Cómo resolver el problema?

Algoritmos

- Descomponer el problema en sub-problemas:
 - Calcular Potencia
 - Calcular Factorial
 - Calcular el i -ésimo Término usando *Potencia* y *Factorial*.



Dra. Jessica Andrea Carballido
CONICET - DCIC (UNS)



¿Cómo resolver el problema?

Algoritmos

ALGORITMO Potencia

DATOS DE ENTRADA: base, Exponente {naturales}

DATOS DE SALIDA: Potencia {natural}

DATOS AUXILIARES:

COMIENZO

Potencia \leftarrow 1

REPETIR Exponente VECES

Potencia \leftarrow Potencia*base

FIN ALGORITMO



Dra. Jessica Andrea Carballido

CONICET - DCIC (UNS)



¿Cómo resolver el problema?

ALGORITMO Factorial

DATOS DE ENTRADA: N {naturales}

DATOS DE SALIDA: Factorial {natural}

COMIENZO

Factorial $\leftarrow 1$

para i desde 1 hasta N

Factorial \leftarrow Factorial $\ast i$

FIN ALGORITMO



Dra. Jessica Andrea Carballido

CONICET - DCIC (UNS)



¿Cómo resolver el problema?

Algoritmos

Ahora utilizo los algoritmos anteriores para resolver el problema de calcular el i -ésimo término.

ALGORITMO Término- i -ésimo

DATOS DE ENTRADA: i {natural}

DATOS DE SALIDA: Término- i -ésimo {real}

COMIENZO

Término- i -ésimo \leftarrow Potencia(2, i)/Factorial(i)

FIN ALGORITMO



Dra. Jessica Andrea Carballido
CONICET - DCIC (UNS)



Si ahora quisiera escribir un algoritmo para calcular la sumatoria de los primeros K términos de la sucesión anterior...

ALGORITMO Suma K Términos

DATOS DE ENTRADA: K {natural}

DATOS DE SALIDA: Sumatoria {real}

DATOS AUXILIARES: i {natural}

COMIENZO

Sumatoria $\leftarrow 0$

PARA i Desde 1 hasta K hacer

Sumatoria \leftarrow Sumatoria + Termino- i -esimo(i)

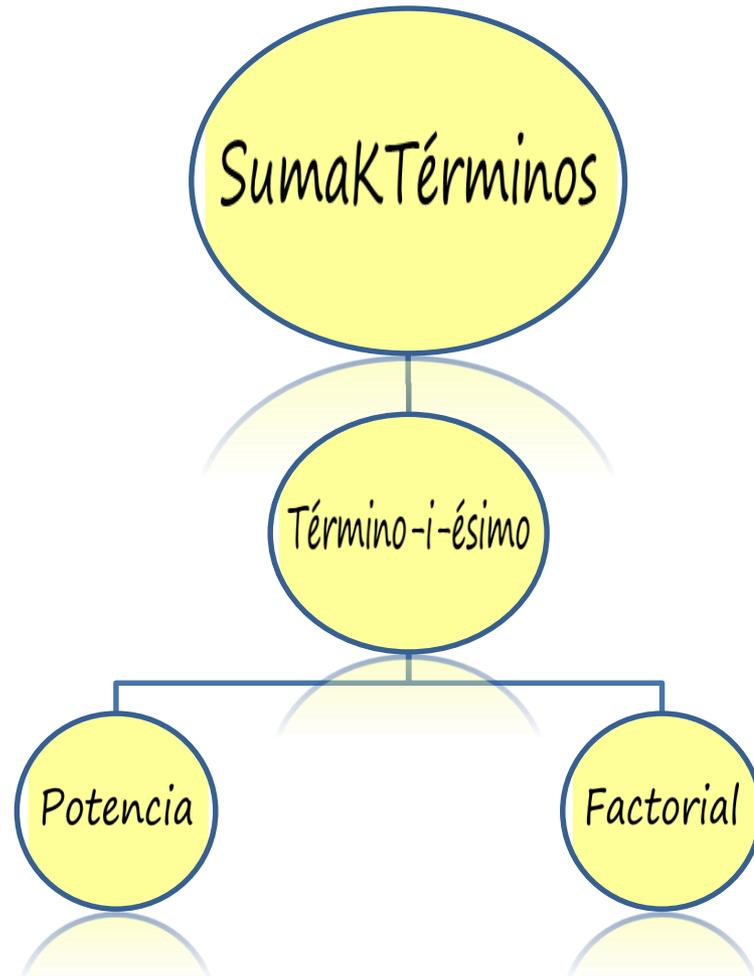
Fin PARA

FIN ALGORITMO

Invocación



Descomposición del Problema



Algoritmos como Primitivas

Un algoritmo se identifica por su **nombre**, sus **datos de entrada** y sus **datos de salida**.

Un algoritmo A puede usar otro algoritmo B como primitiva, y para ello debe indicar:

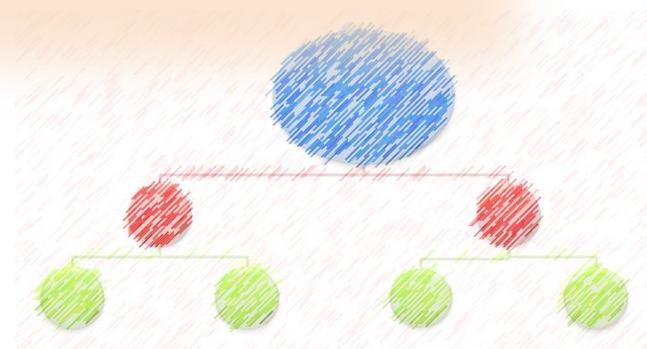
- el nombre de B,
- los datos de A que serán copiados en los datos de entrada para B.

Al usar un algoritmo como primitiva, decimos que lo estamos “invocando” o “llamando”.

Invocación

La invocación de un algoritmo debe coincidir con la definición del algoritmo en los siguientes aspectos:

- ↗ el nombre,
- ↗ la cantidad de datos de entrada,
- ↗ el orden de los datos de entrada,
- ↗ el tipo de los datos (numérico o lógico).



Invocación

Cuando un algoritmo tiene **un sólo dato de salida**, puede ser invocado desde una expresión.

En estos casos, la invocación contendrá el nombre de la primitiva y los datos de entrada.

NOMBRE (dato-ent1, dato-ent2,...)

En cada llamada, los datos de entrada deben tener un valor ya asignado, y ese valor es enviado al algoritmo llamado.

El dato de salida será un único valor devuelto al lugar donde se realizó la invocación.



Ejemplos de invocación

Valor \leftarrow (CantidadDígitos(Num)*2) +1

El dato de salida es único y de tipo entero

SI EsPrimo(Número) = verdadero
ENTONCES...

El dato de salida es único y de tipo lógico

M \leftarrow Máximo3Números(Num1,Num2,Num3)



Para resolver solos
y consultar las dudas

Problema Propuesto

Dados los valores para n , a y b , calcular la siguiente sumatoria:

$$\sum_{i=0}^n (i * a! * ab)$$



Dra. Jessica Andrea Carballido
CONICET - DCIC (UNS)



Problema Propuesto

Calcular cual es el dígito que más veces se repite en un número. Si son varios los dígitos que se repiten más veces, entonces el algoritmo debe devolver el más chico de todos.



- Asuma que cuenta con la primitiva **ContarVeces** vista previamente que cuenta cuantas veces aparece un dígito en un número.



Problemas:

- ✓ Calcular la suma de los dígitos de N . -- RESUELTO --
- ✓ Calcular el promedio de los dígitos de N . -- RESUELTO --

Usando los anteriores como primitiva cuando sea necesario:

- ✓ Sumar todos los números en el intervalo (a, b) cuyo promedio de dígitos sea par.
- ✓ Obtener el promedio de los números contenidos en el intervalo $(a, b]$ cuya suma de dígitos esté entre 5 y 10.
- ✓ Determinar si en el intervalo $[a, b]$ hay algún número que cumpla con esta propiedad: la suma de sus dígitos divide al propio número (por ejemplo el 18, la suma de sus dígitos es 9 y el 9 divide al 18).



Dra. Jessica Andrea Carballido
CONICET - DCIC (UNS)





Muchas
Gracias