















xcvbnmqwertyuiopasdfghjkl  
zxcvbnmqwwertyuiopasdfgh  
jklzxcvbnmqwertyuiopasdfg  
hijklzxcvbnmqwertyuiopasdf  
ghijklzxcvbnmqwertyuiopasd  
fghijklzxcvbnm

# TEORIA DE LA COMPUTACION

## PRACTICA UNO

17/02/2011

INSTITUTO SUPERIOR DE SAN MARTIN PUEBLA

# TEORÍA DE LA COMPUTACIÓN



## PRACTICA DE LA PRIMERA UNIDAD “TEORIA DE LA COMPUTACION”

### INGENIERIA EN SISTEMAS COMPUTACIONALES

“4 semestres”

PROFESORA: yesenia Pérez Reyes

#### ALUMNOS:

Miguel Ángel Cinto Almonte  
Javier Cocoltzi Cuamatzi  
Saúl Morales rodríguez  
Carlos Manuel Morales Moreno

## Índice

En esta parte se da Desde el punto de vista computacional, es necesario disponer de alguna forma de comparar una solución algorítmica con otra, para conocer cómo se comportarán cuando las implementemos,

### Trabajo

Es un concepto que fundamental para todos los programadores, pero sin embargo, a menudo se desconoce por completo.

A cada elemento de esta sucesión se le llama número de Fibonacci.

En lógica matemática y computación, las funciones recursivas o también conocidas como funciones recursivas- $\mu$  son una clase de funciones de los números naturales en los números naturales que son «computables» en un sentido intuitivo. De hecho, en teoría de la compatibilidad se demuestra que las funciones recursivas son precisamente las funciones que pueden ser calculadas con el formalismo, Las funciones recursivas están relacionadas con las funciones primitivas recursivas y su

definición inductiva se construye basándose en la de las funciones primitivas recursivas (estas se obtienen por medio de recursión primitiva y composición de funciones iniciales).

## FACTORIAL

### FACTORIAL EXPLICITAMENTE EN MATEMATICAS

$$M = \{ \mathbf{1*2*3*4*5} \}$$

GRAFICAMENTE

|   |     |
|---|-----|
| 5 | 5   |
| 4 | 20  |
| 3 | 60  |
| 2 | 120 |
| 1 | 120 |

### FUNCION SUMATORIA

$$\Sigma = \{ \textcolor{blue}{1+2+3+4+5} \}$$

GRAFICAMENTE

## CUESTIONARIO

¿Qué diferencias de crecimiento notaste en cada una de las funciones recursivas?

Bueno dependiendo el algoritmo y también el problema

Por ejemplo

Factorial  $\{5!\}=120$

Fibonacci  $\{F_5\}=8$

Sumatoria  $\{\Sigma\}=15$

Factoracci  $\{fab\}=22$

¿Qué pasa cuando hay pocos datos y cuando hay muchos datos?

\*En algunos casos cuando el dato es pequeño el cálculo es casi inmediato y cuando es largo el dato tiene un finito de respuestas

¿Se cumple en todos los casos que en cuanto los datos crecen, la complejidad también crece?

\*si por que también los calculo en una ocasiones vuelve otra ala base y vuelve hacer la operación no solo una vez si no varias veces haciendo mayor complejidad el

algoritmo

¿Existe repetición entre los datos, o todos son diferentes para cada una de las sucesiones?

\*Todas son diferentes por que el algoritmo ataca diferentes problemas no al mismo y diferentes son las instrucciones para atacar el problema

## Conclusiones

Bueno comprendimos la importancia de cómo plantear un algoritmo que para todo problema tiene una solución específica empezando por lo real (como lo haríamos en el método tradicional) de ahí partimos en lo matemático analizar es muy importante comprender antes de resolver y es a veces nos falta la capacidad de analizar un problema Atacar el problema específico recordemos que si ya utilizamos el algoritmo no podemos utilizarlo para resolver otro problema es por eso que debemos realizar otra estrategia de hecho nosotros utilizamos algoritmos

toda la vida para hacer una actividad específica y no es repetitivo verdad así hacemos con un problema matemático imaginemos hacer una multiplicación que quiero sacar el número 6

Mi algoritmo

Seria agarrar 2 números cuales fueran

$$\{2\} * \{3\} = 6$$

No puedo usarlo para resolver una división

$$\{2\} \% \{3\} = 0.66$$

Conclusión no llego a lo que yo realmente necesito