TRABAJO NÚMERO 1

COMPLEJIDAD COMPUTACIONAL

PRESENTADO POR

CARLOS ARTURO MORENO TABARES

PRESENTADO A

CARLOS AUGUSTO MENESES ESCOBAR

UNIVERSIDAD TÉCNOLOGICA DE PEREIRA

FACULTAD DE INGENIERIAS

MAESTRIA DE INGENIERIA DE SISTEMAS Y COMPUTACIÓN

PEREIRA

2019

TRABAJO NRO 1: **Complejidad en algoritmos simples**

1. Desarrollar una función tal que reciba dos números enteros positivos y devuelva el valor del mcm (mínimo común múltiplo). Se debe desarrollar el algoritmo usando ciclos y otra solución con el método de Euclides. En cada caso se debe establecer cuál es la complejidad algorítmica y ¿por qué?
   1. Mínimo común múltiplo con ciclos

Una manera sencilla de realizarlo a *“fuerza bruta”* es ir multiplicando los valores entrantes por un factor que empiece desde 1 para ambos el cual en cada iteración verifica:

Si eso se cumple es el mínimo común múltiplo, pero en caso tal que no sean iguales debe determinar quien es mayor, es decir:

Si resulta ser verdadero entonces el factor que debe aumentar en 1 más es el factor de Y, de lo contrario será al factor de X y luego volver a verificar la igualdad, y así hasta encontrar el “común”, el código fuente en lenguaje **Python** que satisface dicha situación es el siguiente:

# O(n) where the max n is x+y (getting with hard mode)

def mcm\_cycles(x, y):

    if(x<=0 or y<=0):               # O(1)

        return 0                    # O(1)

    # factor control for multiply

    factorX=1                       # O(1)

    factorY=1                       # O(1)

    while True:                     # O(x+y) -> number of max iterations,

                                    # iter per x factor and y factor to

                                    # get an equal multiply factor result

        if(x\*factorX==y\*factorY):   # O(1) -> factor multiply operation

                                    # is equal between numbers

            # return a mcm

            return x\*factorX        # O(1)

        if(x\*factorX>y\*factorY):    # O(1)

            factorY+=1              # O(1)

        else:                       # O(1)

            factorX+=1              # O(1)

En términos de la matemática el máximo de “pasos” que lograrían entre estos números es ya que una vez alcance el factor de multiplicación se abra dado un “ciclo” completo de la multiplicación porque una vez que el factor multiplicativo continua se comportan igual pero en un “siguiente” nivel por lo tanto el máximo de iteraciones que puede lograr es la suma de los valores entrantes entre si donde estos valores para la función anterior es ‘x’ y ‘y’; este comportamiento se puede observar en la siguiente gráfica.

La grafica anterior es de tiempos de ejecución la estrategia que se aplicó para las ejecuciones es un rango de los números de Fibonacci desde el número de la sucesión 20 hasta el 40, y la ejecución se realizó con x>y para garantizar que no se repitan números entre sí; en el tiempo de ejecución de la función puede indicar de manera directamente proporcional la cantidad de iteraciones, es decir que entre más tiempo de ejecución más cantidad de ciclos, al correr los valores con mayor tiempo de ejecución y contabilizar los ciclos su cantidad de iteraciones no superaba veces, por lo cual se concluye que la función su complejidad es **Lineal** o

* 1. Mínimo común múltiplo con el mcd de Euclides

El algoritmo de Euclides dice que el máximo común divisor de los números ‘x’ y ‘y’ es la operación

Si el el máximo común divisor es ‘y’, en caso de que no sea el entonces debe aplicarse:

Si el el máximo común divisor es , en caso de que no sea el entonces debe repetirse el algoritmo de la siguiente manera:

Se repite hasta que el sea igual a 0 y en el momento que lo sea el mcd es o bien el dividendo del momento.

**Fuente:** <http://aprendeenlinea.udea.edu.co/boa/contenidos.php/8b077438024e1bddfbc83706da8049f2/138/1/contenido/contenido/alg_euclides.html>

Con el máximo común divisor se puede obtener el minimo común múltiplo por medio de la siguiente propiedad:

Por tanto si se tiene los valores de ‘x’, ‘y’ y del , se debe despejar el mcm de la propiedad quedando de la siguiente manera:

**Fuente:** <https://thales.cica.es/rd/Recursos/rd98/Matematicas/07/mcdmcm.html>

Teniendo en cuenta lo anterior al codificar el algoritmo en lenguaje **Python** queda de la siguiente manera:

def mcd\_euclides(divider, dividend):

    while True:                 # Max of iterations is number of

                                # divisions where complexity is O(n)

                                # n is 4,4 \* 10\*\*-9 percentaje from

                                # dividier + dividend

        # get mod x/y, next cases is y/modR\_1, modR\_1/modR\_2 ...

        # modR\_n/modR\_n+1

        modR=divider%dividend   # O(1)

        if(modR==0):            # O(1)

            return dividend     # O(1)

        divider=dividend        # O(1)

        dividend=modR           # O(1)

def mcm\_with\_mcd(x,y,mcd):

    return (x\*y)/mcd        # O(1)

En términos de *“pasos”* o iteraciones de este algoritmo es dependiente de las divisiones que se efectúan, al mirarse cuidadosamente cada 2 divisiones logra reducir un 25% del total de iteraciones máxima que **había** en el algoritmo anterior con ciclos de factores, para verificarlo se ejecutó con un rango amplio de números de la sucesión de Fibonacci desde el 20 hasta el 100 (ya que muestra ser más efectivo) dando la siguiente grafica

El control de numeración en el eje ‘x’ se aplicó la misma estrategia utilizada en el anterior (‘x’ con punto decima ‘y’), se logra notar que para ese rango de valores la ejecución es mucho más rápida, se consideró los más demorados donde más iteraciones se realizaron y se determino que para el el máximo de iteraciones posibles con el algoritmo anterior que es 225851434150 con el algoritmo de Euclides realizó 10 y 10 iteraciones para 225851434150 es el es decir es millones de veces más efectivo que con ciclos utilizando factores, sin embargo la complejidad no deja de ser **Lineal** o , ya que de todas formas si se utilizara en números infinitos el valor de iteraciones aumentará (muy poco tiempo de ejecución).

**Fuente:** <https://www.iteramos.com/pregunta/81246/el-tiempo-de-complejidad-del-algoritmo-de-euclides>

1. Utilizando el método por Inducción, resolver los siguientes ejercicios:
   1. Pruebe que 3 es divisor de para todo entero positivo **n**

**Caso base:**

Si 3 es divisor de , es porque la operación de división al reemplazar n es un número entero, como caso base será n=1

Al reemplazar y realizar la operación como resultado se tiene el valor 1 que es entero, permitiendo continuar al paso inductivo.

**Paso inductivo:**

En el paso inductivo se plantea como verdad el caso de que 3 es divisor de todo número entero positivo k para la siguiente formula

Se plantea como **hipótesis** de que es cierto y luego se presente demostrar que para el siguiente también cumple a la propiedad

# Demostración:

La demostración indicará que de la fórmula para todo número entero positivo resultará un número entero

Hipótesis

Demostrar divisible por 3

Inicio:

La forma que mostrará que el resultado es entero es que la formula dada será divida toda con 3

Resolver multiplicaciones

Resolver restantes y ordenar

Factor común 3

Por cierta propiedad de los fraccionarios puede separarse con un denominador homogéneo dejando en medio una operación suma () de la siguiente manera

Luego se simplifica

La parte de la operación donde está resultará un número entero ya que es igual a la hipótesis planteada en el paso inductivo la cual resultará, mientras que a la otra parte son suma de enteros es decir:

La suma de números enteros con otros números enteros da como resultado otro número entero lo cual **concluye** que si es divisible entre 3

* 1. Pruebe si **n** es un entero positivo cualquiera, entonces se cumple la siguiente fórmula para la suma de cubos

**Caso base:**

Si la suma de cubos cumple con la formula dada, es porque la suma uno a uno su resultado será igual a la formula, se comprobará para n=3

Suma uno a uno:

Suma con la fórmula:

La fórmula cumple para el caso de n=3, permitiendo continuar al paso inductivo

**Paso inductivo:**

En este paso se plantea como verdadero que la formula efectúa correctamente la sumatoria de cubos de 1 hasta n, será planteado todos los números enteros positivos como k, quedando la fórmula de la siguiente manera:

De esta manera queda como **hipótesis** para apoyar en la demostración

**Demostración:**

La demostración indicará que la sumatoria de cubos cumple para todo valor k, si cumle deberá cumplir también para el siguiente valor que es k+1

Hipótesis

Demostrar

Se inicia de la formula a demostrar

Resolviendo la potencia

Se aplica distributiva

Es separado el fraccionario por sumas homogéneas

Factor común 4

Se simplifica el 4

Ahora es solucionada la multiplicación por suma de exponentes (tienen igual base)

Finalmente se cambia de lado de la igualdad el valor

Llegando de esta manera a la igualdad de la hipótesis, lo cual concluye que la formula si cumple para la sumatoria de cubos.

* 1. El juego de Nim se juega entre dos personas con las siguientes reglas: Se pone un número de n fichas iguales sobre la mesa. Cada jugador en su turno puede tomar 1, 2 o 3 fichas. El jugador que toma la última ficha pierde. Demuestre que el primer jugador tiene una estrategia ganadora siempre y cuando

Primero se debe de comprender el *“juego”* que como mínimo debe tomar 1 ficha, es decir debe forzarse que llegue a un momento en donde el adversario quede con 1 sola ficha, para ello debe asegurarse que en cada turno del adversario tenga múltiplo de 4 +1 de esa manera no podrá colocar al jugador principal con 1 ficha al final ya que si el otro jugador saca 1 ficha el jugador principal sacará 3, si otro jugador saca 2, entonces jugador principal saca 2 si el otro jugador saca 3 fichas el jugador principal saca 1 dando así siempre 4 por lo cual quedará siempre 1 al final, lo anterior cumple la siguiente formula:

Donde n es el total de fichas y k es un entero positivo, con ello se puede realizar el proceso de inducción.

**Caso base:**

En esta etapa se verificará que la formula cumple lo mencionado, se probará con un valor determinado en este caso con .

Es decir que el jugador que en su turno empiece con 9 fichas ya perdió porque:

* Si en el turno del jugador principal hay 9 fichas si juega 1 ficha quedarán 8 fichas para que el otro jugador gane debe sacar 3 fichas para que queden 5, de esa forma el jugador principal en su último turno quedará con 1 ficha.
* Si en el turno del jugador principal hay 9 fichas si juega 2 fichas quedarán 7 fichas para que el otro jugador gane debe sacar 2 fichas para que queden 5, de esa forma el jugador principal en su último turno quedará con 1 ficha.
* Si en el turno del jugador principal hay 9 fichas si juega 3 fichas quedarán 6 fichas para que el otro jugador gane debe sacar 1 ficha para que queden 5, de esa forma el jugador principal en su último turno quedará con 1 ficha.

Por lo tanto, cumple para todas las jugadas posibles que perderá, permitiendo continuar al paso inductivo.

**Paso inductivo:**

En el paso inductivo se asumirá que para todo número entero k, los resultados de la siguiente formula:

Se plantea la **hipótesis** que, si el jugador que tenga en uno de sus turnos el valor resultante en su último turno tendrá 1 sola ficha, es decir que perderá si el segundo jugador sostiene para cada turno un valor de la formula en su adversario.

**Demostración:**

Se demostrará que si en el turno de uno de los jugadores es de cierta cantidad ese jugador siempre perderá si se sostiene en esos valores en cada que sea su turno y que cumple para el valor de k siguiente es decir k+1.

Hipótesis

Demostración

Iniciar desde la formula a demostrar

Resolver la distributiva

Ajustar el orden para visualizar la hipótesis

Como se puede observar a la demostración se llegó que se tiene la hipótesis sumando un valor 4 se sigue cumpliendo de valores de dicha cantidad ya que al sumar 4 al resultado de la hipótesis es como si fuese k multiplicado por k mayor, dando otro de los números para “perder”, se cumple por lo que no hay forma de sacar 4 fichas, ahora bien si se inicia con un valor diferente al de la formula se puede forzar a que el adversario tenga en su turno un valor que cumple la formula; ya que solo se puede retirar fichas menores a 4, es decir que si se hace la demostración para los otros casos se tiene que:

Demostración

Iniciar desde la formula a demostrar

Resolver la distributiva

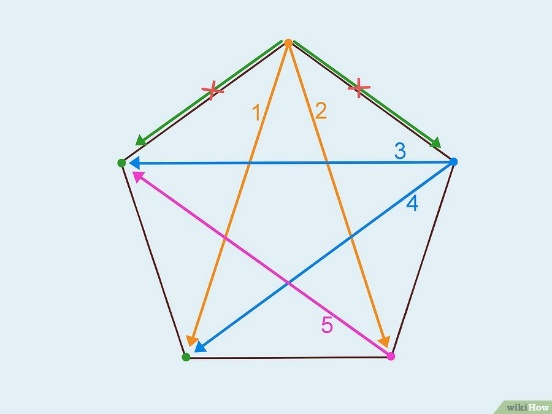
Ajustar el orden y descomponer la suma para visualizar la hipótesis

En este caso si se tiene una cantidad como resultó al final de la anterior demostración si se quita 1 ficha, el otro jugador tendrá la cantidad para perder, ya que vuelve a la conclusión ya dada, esto aplica respectivamente para 3 y 4.

* 1. Pruebe que el número total de diagonales que tiene un polígono convexo de n lados , es

Las diagonales de un polígono de manera visual se obtienen trazando rectas desde un vértice (punto del polígono) a otro vértice contará +1 de diagonales si esta no queda sobre-puesta a los lados del polígono, así como se muestra en la siguiente imagen:

Figure 1 Diagonales de un pentágono



Fuente: <https://es.wikihow.com/calcular-cu%C3%A1ntas-diagonales-tiene-un-pol%C3%ADgono>

Teniendo en cuenta el cálculo de diagonales cumple a una sucesión 0, 2, 4, 5, 9, 14 se puede proceder al proceso inductivo.

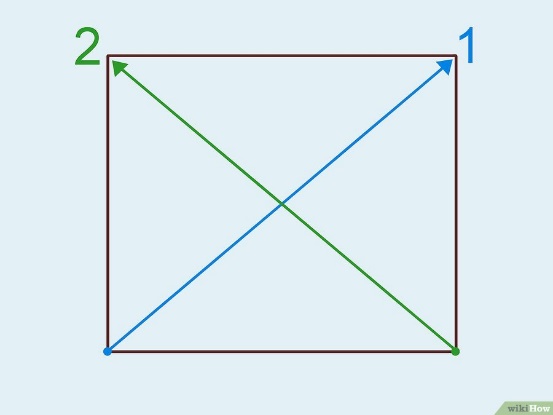
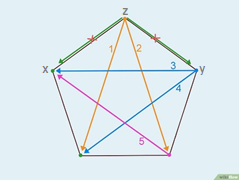
**Paso inductivo:**

Se plantea como **hipótesis** que la fórmula para obtener la cantidad de diagonales es correcta y por convención se cambia los valores n por k quedando de la siguiente manera:

Se debe tener muy presente que k es la cantidad de lados y n el total de diagonales, se procede a preparar la demostración.

## Demostración:

Ahora se demostrará que el valor de diagonales corresponde a la fórmula para el k siguiente, teniendo en cuenta que el valor k es la cantidad de lados, si aumentara la cantidad de lados en 1 más en el caso del cuadrado al pentágono seria

Si visualizamos el vértice ‘z’ la cantidad de diagonales que se pueden trazar son 2 si tenemos en cuenta sus lados seria (donde n son sus lados), si se fuera a agregar un lado más seria al reemplazarlo en n seria , pero también debe incluirse el cuadrado que formaría entre los vértices por lo cual la cantidad de diagonales para un lado más es (viéndolo desde los vértices ‘x’ y ‘y’)

Hipótesis

Demostrar

Se inicia desde la formula a demostrar con la igualdad respectiva al caso

Se resuelve la parte derecha de la igualdad

Aplicando distributiva

Acomodando y ordenando los valores para factorizar y ver la hipotesis

Separando fraccionarios por suma de denominadores homogeneos

Factorizando para simplificar

Simplificando

Cambiando de lado de desigualdad al valor

La igualdad llegó al valor de la hipótesis

Por lo cual se demuestra que la formula encuentra para todo entero positivo k la cantidad de diagonales de un polígono. <http://rinconmatematico.com/foros/index.php?topic=49108.0>

* 1. Pruebe que para todo entero positivo n:

**Caso base:**

En el caso base se verificará que cumple para un valor particular, en este caso para seria:

Para facilitar la suma se amplifican cada uno de los fraccionarios al minimo común múltiplo de los denominadores en este caso es 60

Se simplifica el fraccionario de la izquierda de la desigualdad a su mínima expresión con su máximo común divisor el cual es 12

Lo cual cumple, permitiendo continuar al paso inductivo.

**Paso inductivo:**

Se plantea como **hipótesis** que para todo entero positivo k es correcto para la siguiente formula

Con esta hipótesis será planteado la demostración.

**Demostración:**

Se quiere demostrar que cierta formula cumple una sumatoria para todo entero positivo k, para ello se demuestra que cumple para el k siguiente, lo cual es:

Hipótesis

Demostrar

Partiendo desde lo que se va demostrar

Se multiplica numerador y denominador por (k+1) para alcanzar el valor del otro lado de la desigualdad

Resolviendo distributiva del numerador

Es separado por suma fraccionarios con denominador homogéneo al lado derecho de la desigualdad

Factorizando la expresión

Simplificando el fraccionario con

Cambiando del lado de la igualdad al valor

Queda como resultado la hipótesis

Por lo cual se concluye que la formula se cumple respectivamente para todo entero positivo.

* 1. Considere la sucesión de Fibonacci y demuestre que

La sucesión de Fibonacci es la suma del valor actual con el anterior, los primeros 2 valores de Fibonacci son 1.

**Caso base:**

En este paso se verificará si la formula cumple lo que indica, será comprobado con n=3

Equivalencia de los Fibonacci de 1 hasta 4

Reemplazando respectivamente

Resolviendo potencias

Ambos lados de la igualdad cumplen correctamente, permitiendo proceder al paso inductivo.

**Paso inductivo:**

Se plantea la **hipótesis** de que la formula con dicha sucesión cumple respectivamente y adicionalmente se modifica por convención la n por k,

Se debe de tener presente que el Fibonacci es la suma su valor actual (o más bien justo anterior con el otro anterior) de la sucesión con el anterior es decir

**Demostración:**

Es planteado en la demostración que la formula dada con dicha sucesión cumple para el k siguiente, tomando como base la hipótesis, resultando el siguiente planteamiento

Hipótesis

Demostración

Partiendo de la formula a demostrar

Reemplazar el Fibonacci de la parte derecha de la igualdad (a su equivalente según la serie de Fibonacci

Aplicando distributiva

Es reemplazado el Fibonacci de la parte derecha de la igualdad a su equivalente según la serie de Fibonacci

Para notar la igualdad se efectuará una vez más el reemplazo del Fibonacci equivalente a

Se cambia del lado de la igualdad a

Es decir que siempre reduciendo resultará el k anterior para la sumatoria de Fibonacci al cuadrado hasta que la iteración multiplicativa será en algún momento

Lo cual es 1\*1 es decir que cumple lo que se ha llevado de iteraciones hasta el momento es para k=3

Debido a cómo funciona la serie se aplica quedando lo siguiente:

Cabe recordar que en la serie el valor es 1 por lo cual se puede afirmar que de esta manera se demuestra que si cumple para todo k entero positivo.

* 1. Pruebe que cumple para todo entero n positivo.

Esta es una desigualdad donde debe cumplirse que el lado izquierdo es mayor al derecho con el valor n que se ponga primero se ha de verificar que cumple para un caso específico.

# Caso base:

Se verificará si la desigualdad cumple para n=3

El factorial es la multiplicación del rango de números enteros desde 1 hasta el valor n, es decir que

Con n=3 cumple a la desigualdad permitiendo a proceder con el paso inductivo.

**Paso inductivo:**

Se plantea la **hipótesis** de que la desigualdad cumple para los números enteros positivos hasta su variable n, pero por convención la n será reemplazada por k donde cumple de todos los enteros positivos hasta k.

**Demostración:**

Para demostrar que la formula cumple para todo número entero positivo se demostrará que cumple también para el k siguiente es decir al k+1 quedando asi:

Hipótesis

Demostrar

Se empieza desde la desigualdad a demostrar:

Resolviendo en el exponente de la parte derecha de la desigualdad

Reordenando para ajustarse a la hipótesis

Se separa por multiplicación de exponentes de igual base

Ahora se descompone la parte interna del factorial

Se aplica la propiedad del factorial

Se puede factorizar

Se reordena para detallar la hipótesis

De acuerdo a la hipótesis ya cumple, por lo cual si alguno de estos valores se les multiplica un valor diferente a cero siempre será el mismo valor o superior, y en como va el proceso de la demostración se mostrará que el lado izquierdo que contiene aun valores k es mayor al valor

Se probará para k=1

Al cumplirse, para 1, para todos los demás valores cumple ya que el lado derecho de la desigualdad nunca cambia y el lado izquierdo siempre aumentará y además la multiplicación de un número positivo con otro número positivo siempre resultará un valor mayor