**UNIVERSIDAD DEL NORTE**

DIVISIÓN DE INGENIERÍAS

DEPARTAMENTO DE SISTEMAS

CÓDIGO: 200055832

NOMBRE:Luis Sebastian Caicedo pimienta , FCHA: 11 / 02 / 2016

DD MM AAAA

Práctica No. 04

1. Diseñar la función de tiempo T(¿?) y concluya el orden de complejidad de la siguiente función suponiendo que el valor de entrada n es par, potencia de dos y n>0:

Potencia(n)

i←0 // i, n son definidos como enteros

Mq (n%2 = 0) haga //% Es la función módulo dos.

n←n/2

i←i+1

Fin\_Mq

Retorne i

Fin

R/.

Si n es impar 🡪 el numero no es potencia de 2 entonces T(n)= k

Si n es par🡪 T(n)= O(Log2(n))

1. Demostrar la regla de la multiplicación de funciones asintóticas: T1(n) \* T2(n) es O(f(n) \* g(n))

R/. Dadas las constantes c1, c2, c3 , n1 y n2,

Si n ≥ n1, entonces T1(n) ≤ c1\*f(n), y si n ≥ n2 , entonces T2(n) ≤ c2\*g(n). Sea n0 = máx (n1, n2).

Si n ≥ n0, T1(n) \* T2(n) ≤ c1\*f(n)\* c2\*g(n) ≤ c3\* (f(n)\*g(n)), siendo c3 ≥ 0 y c3 = c1 \* c2.

Luego, aplicando la regla de la constante :

T1(n) \* T2(n) ≤ O(f(n) \* g(n)); por lo tanto: T1(n) \* T2(n) es O(f(n) \* g(n)).

1. Si el tiempo de ejecución de un programa en función del número de datos de entrada es igual a , considerando como unidad del modelo el segundo, resolver las siguientes preguntas: i) Si se dispone de un tiempo de 1 hora qué tamaño de problema se puede resolver. ii) Suponga que el tiempo se dispone es de 1000 Seg, qué tamaño de problema se puede resolver?. iii) Suponga que se tiene una máquina 10 veces más rápida; para un tiempo de 1000 Seg, ¿qué tamaño de problema se puede resolver?. iv) Suponga que el tiempo de ejecución de un algoritmo es igual a , si se tiene un tiempo de 1 millón se segundos ¿Qué tamaño de problema se puede resolver?, justifique la respuesta del ¿porqué da dicho tamaño de problema?.

R/.

i) Al disponer un tiempo de 1 hora, entonces:

ii) Como se dispone de un tiempo igual a 1000 segundos, entonces:

iii) Como se tiene una máquina 10 veces más rápida, para un tiempo 1000 segundos entonces

iv) Como el tiempo es de 1 millón de segundos, para un programa con tiempo de ejecución igual a , entonces:

La explicación de que el tamaño de problema es pequeño, es porque el tiempo de demora del programa es exponencial.

1. Suponga que el modelo de un computador trabaja en µ\_Seg , y el tiempo de desempeño de dos algoritmos en función del número de datos de entrada son ; para un valor de , seleccione el mejor algoritmo justificando la respuesta.

R/.

1. Calcular prácticamente el orden de complejidad O(¿?) de un algoritmos, cuyo tiempo de desempeño es igual a , concluyendo la respuesta en términos de la incidencia de la constante en la complejidad del algoritmo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Orden de Complejidad |
| 1 | 100 | 0,1 | 1.000 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Comprobar matemáticamente que

R./ Por definición matemática el valor de

1. Dadas dos funciones

Comprobar prácticamente si o Si

R/. Se le dan valores a los datos de entrada ( y a las funciones

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Luego se puede concluir que:

1. Diligenciar la siguiente tabla identificando la clase de complejidad asociada a cada problema Π

|  |  |  |  |
| --- | --- | --- | --- |
| Π |  |  |  |
| P=Polinomial | Alg. Det y O(P(n)) | Proc:  Lea n  Para i=1,n,1  Para j=1,n,1 haga  A(i,j)←0  Fin\_para  Fin\_para  Para i=1,n,1 haga  A(i,i)←1  Fin\_para  Fin\_proc | Sea T(n) el tiempo de de |
| E=Exponencial | Alg \_\_\_\_\_\_\_\_  y  O( | Fib(n)  Si (n=1)o(n=0) entonces  Num\_Fib←n  Si\_no  Num\_Fin←Fib(n-2)+Fib(n-1)  Fin\_si  Fin\_función\_Fib | Es un algoritmo tipo |
| NP | Alg. \_\_\_\_\_\_\_  y  O(P(n)) | Search(L,n,x)  i←1  Mq(x < > L(i) and i<=n) haga  i←i+1  Fin\_Mq  Si (x=L(i)) entonces  Devolver i  Si\_no  Devolver 0  Fin\_si  Fin\_Search | Es un Alg  Luego |
| T=Tratables | Sol, \_\_\_\_\_\_\_  en  Tiempo y \_\_\_\_\_\_\_ | Asignador de llaves de longitud L no repetidas en una función de Hash |  |
| I=Intratables | Sol. \_\_\_\_\_\_\_\_\_  en  \_\_\_\_\_\_ y \_\_\_\_\_\_\_ | Función Exposec(a,n)  // Para números (a > 1 billón digs).  r←a  Para i=1,n-1,1 haga  r←a\*r  Fin\_para  Devolver r  Fin\_Exposec |  |
| NP\_C  NP\_H | No se han encontrado Alg., polinomiales para su solución pero no se han demostrado que no existen | VP(n)  Dado un entero positivo n decidir si n es o no primo |  |
| U=Undecible | No\_\_\_\_\_ | Traductor de lenguajes y dialectos multilingües escrito y hablado 100% |  |

1. Diligenciar la siguiente tabla identificando la clase de complejidad P, E, NP, NP\_C o U asociada a cada problema Π

|  |  |  |  |
| --- | --- | --- | --- |
| Problema Π | Clase de Complejidad | T(¿?), O(¿?) | Justificación |
| FAC(n) dado un antero positivo n encontrar sus divisores |  |  |  |
| **Productor (A,B,C,n)**  Para i=1 hasta n con incrementos de 1 haga  Para j=1 hasta n con incrementos de 1 haga  C(i,j) ← 0  Fin\_para  Fin\_para  Para i=1 hasta n con incrementos de 1 haga  Para j=1 hasta n con incrementos de 1 haga  Para k = 1 hasta n con incrementos de 1 haga  C(i,j) ← C(i,j) + (A(i,k) + B(k,j))  Fin\_para  Fin\_para  Fin\_para  **Fin\_Productor** |  |  |  |
| Calcular la satisfactibilidad (condición de verdad) de una fórmula lógica:  \*) |  |  |  |
| Colorado de grafos (CG) |  |  |  |
| Verificador automático de satisfactibilidad de algoritmos |  |  |  |
| Proc Inner Loop Selection Sort (A,n,i)  Small←i  Para j=i+1,n,1  Si (A(j) < A(small)) entonces  Small←i  Fin\_para  Fin\_para  Fin\_proc |  |  |  |
| **Determinante (M,d)** // Determinante d matriz escalar  d ← 1 ; i ← 1  Mq (i <= n ) haga  d ← d \* M(i,i)  i ← i + 1  Fin\_Mq  Escriba d  **Fin\_Determinante** |  |  |  |
| Paso de objetos físicos a través de la red Internet |  |  |  |
| Problema de empaquetamiento “Bin packing” |  |  |  |
| Cálculo de los caminos de longitud k de costo mínimo de un grafo G(n,v) |  |  |  |

1. Diseñar una estructura de datos minimal y clasificar en su orden de complejidad el tipo de problema Π=Diseñar un simulador de semáforos representativo de la intersecciones de las vías:

f.