

Análisis de algoritmos

Fecha: 29/08/2014

Instituto Tecnológico de Costa Rica

Profesor : Mauricio Rojas Fernández

Grupo 40

Tarea #2

Estudiante Argenes Daniel Montoya Aguilar

Carnet 201106385

1.

Expresión	Termino dominante	O(...)
$5 + 0.001n^3 + 0.025n$	$0.001n^3$	$O(n^3)$
$500n + 100n^{1.5} + 50 \log \log_{10} n$	$100n^{1.5}$	$O(n^{1.5})$
$0.3n + 5n^{1.5} + 2.5 n^{1.75}$	$2.5 n^{1.75}$	$O(n^{1.75})$
$n^2 \log_2 n + n (\log_2 n)^2$	$n^2 \log_2 n$	$O(n^2 \log_2 n)$
$n \log_3 n + n \log_2 n$	$n \log_2 n$	$O(n \log_3 n)$
$3 \log_8 n + \log_2 \log_2 \log_2 n$	$3 \log_8 n$	$O(3 \log_8 n)$
$100n + 0.01n^2$	$0.01n^2$	$O(n^2)$
$0.01n + 100n^2$	$100n^2$	$O(n^2)$
$2n + n^{0.5} + 0.5n^{1.25}$	$0.5n^{1.25}$	$O(n^{1.25})$
$0.01n \log_2 n + n(\log_2 n)^2$	$n(\log_2 n)^2$	$O(n(\log_2 n)^2)$
$100n \log_3 n + n^3 + 100n$	n^3	$O(n^3)$
$0.003 \log_4 n + \log_2 \log_2 n$	$0.003 \log_4 n$	$O(\log 4 n)$

2. Un algoritmo cuadrático tiene un tiempo de procesamiento $T(n) = c * n^2$ y tarda $T(Z)$ segundos para procesar Z elementos de datos. Cuanto tiempo tardaría en procesar $n = 5000$ elementos de datos, asumiendo que $Z = 100$ y que $T(Z) = 1\text{ms}$?

$$T(z) = c * z^2 = 1\text{ms} = c * 100^2 = 1\text{ms}/100^2 = c = 0.0001$$

$$T(n) = c * n^2 = T(5000) = 0.0001 * 5000^2 = 2500 \text{ ms}$$

3. Un algoritmo con una complejidad de tiempo $O(f(n))$ y un tiempo de procesamiento $T(n) = c f(n)$ donde $f(n)$ es una función conocida de n , tarda 10 segundos para procesar 1000 elementos de datos. Cuanto tiempo tomara procesar 100000 elementos de datos si $f(n) = n$ y si $f(n) = n^3$?

$$F(n) = n$$

$$T(n) = c * n$$

$$10 = c * 1000$$

$$10/1000 = c$$

$$C = 0,01$$

$$T(n) = 0,01 * 100000$$

$$T(n)=1000$$

$$F(n)=n^3$$

$$T(n)=c*n^3$$

$$10=c*1000^3$$

$$10/1000^3=c$$

$$C=1 \times 10^{-8}$$

$$T(n)=1 \times 10^{-8} * 100000^3$$

$$T(n)=10000000$$

4. Se tienen dos paquetes de software A y B de complejidad $O(n \log n)$ y $O(n)$ respectivamente. Y se tiene que

$$T_A(n) = C_a n \log_{10} n \text{ y}$$

$$T_B(n) = C_b n$$

expresan el tiempo en milisegundos de cada paquete.

Durante una prueba, el tiempo promedio de procesamiento de $n = 10^4$ elementos de datos con los paquetes A y B es de 100ms y 500ms respectivamente. Establezca las condiciones en las cuales uno de los paquetes empieza a desempeñarse mejor que el otro y recomiendo la mejor opción si se van a procesar $n = 10^9$

$$T_a(n)=100\text{ms}$$

$$T_a(n)=C_a * n \log_{10} n$$

$$100=C_a * 10^4 \log_{10} 10^4$$

$$10/4 \log_{10} 10^4 = 2,5 \times 10^{-3}$$

$$T_b(n)=500\text{ms}$$

$$T_b(n)=C_b * n$$

$$500=C_b * 10^4$$

$$500/10^4=0,05$$

$$0,05n \leq 0,0025n \log_{10} n$$

$$0,05n/0,0025n \leq \log_{10}n$$

$$20 \leq \log_{10}n$$

$$10^{20} \leq n$$

A partir de $n=10^{20}$ $Tb(n)=C_b \cdot n$ empieza a desempeñarse mejor.

5. Asuma que el arreglo a contiene n valores, que el método `randomValue` toma un número constante c de pasos computacionales para producir cada valor de salida, y que el método `goodSort` toma $n \log n$ pasos computacionales para ordenar un arreglo. Determine la complejidad O para el siguiente fragmento de código:

```
for ( i = 0; i < n; i++ ) {
    for ( j = 0; j < n; j++ ) {
        a [ j ] = randomValue ( i );
    }
    goodSort( a );
}
```

$$1 + (3 + (1+3+2+c)n + n \log n)n =$$

$$1 + (3 + (6+c)n + n \log n)n = 1 + (3 + 6n + cn + n \log n)n = 1 + 3n + 6n^2 + cn^2 + n^2 \log n =$$

$$1 + 3n + (6+c)n^2 + n^2 \log n$$

$$\text{Término dominante} = (6+c)n^2$$

$$\text{Complejidad } O = O(n^2)$$

6.

El algoritmo contiene los siguientes pasos:

- ✓ Crear una colección de casilleros vacíos
- ✓ Colocar cada elemento a ordenar en un único casillero
- ✓ Ordenar individualmente cada casillero
- ✓ devolver los elementos de cada casillero concatenados por orden

Pseudocódigo

función bucket-sort(elementos, n)

casilleros ? colección de n listas

para i = 1 hasta longitud(elementos) hacer

c ? buscar el casillero adecuado

insertar elementos[i] en casillero[c]

fin para

para i = 1 hasta n hacer

ordenar(casilleros[i])

fin para

devolver la concatenación de casilleros[1],..., casilleros[n]

Aquí elementos es la lista de datos a ordenar

y n el número de casilleros que queremos usar.

7.

Máquina de turing que reemplaza el Nick por el nombre

Nick: bid

Nombre: argen

Σ (alfabeto de la máquina) = b,i,d,a,r,g,e,n, \square

Γ (alfabeto de la cinta) = b,i,d, \square

Q (conjunto de estados) = q_0, q_1, q_2

q_0 (estado inicial) = q_0

f (función de transición) = $Q \times \Gamma$

F (estado final) = q_2





