

CS023 – Algoritmia y Complejidad

Instrucciones: Resuelva los siguientes ejercicios de forma clara y ordenada, dejando constancia de todo su procedimiento.

Ejercicio 1. Compruebe que $f(n) = n^2 \log n + n$ es $O(n^2 \log n)$, $\Omega(n^2 \log n)$ & $\Theta(n^2 \log n)$.

Ejercicio 2. Compruebe que $f(n) = \log n!$ es $O(n \log n)$ & $\Omega(1)$.

Ejercicio 3. Compare las funciones $f(n) = n^{\log n}$ & $g(n) = 2^{\sqrt{n}}$.

Ejercicio 4. Compare las funciones $f(n) = 2^{\log n}$ & $g(n) = n^{\sqrt{n}}$.

Ejercicio 5. Compare las funciones $f(n) = 2n$ & $g(n) = 3n$.

Ejercicio 6. Dé una estimación en notación *big-oh* para el número de operaciones (comparación o multiplicación) usadas en el siguiente segmento de un algoritmo:

```
a=[a1,a2,...,an]      #array de números
m=0
for (i=1;i<=n;i++){
    for (j=i+1;j<n;j++){
        m=max(a[i]*a[j], m)
    }
}
```

Ejercicio 7. El algoritmo convencional para evaluar un polinomio

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \text{ en } x = c$$

puede expresarse mediante el siguiente pseudocódigo:

```
a=[a0,a1,...,an]      #coeficientes del polinomio
power=1
y=a[0]
for (i=1;i<=n;i++){
    power=power*c
    y=y+a[i]*power
}
```

- Evalúe $3x^2 + x + 1$ en $x = 2$ usando el algoritmo descrito.
- Dé una estimación en notación *big-oh* para el número de operaciones (suma o multiplicación) usadas en el algoritmo.

Ejercicio 8. Hay un algoritmo más eficiente (en términos del número de sumas y multiplicaciones usadas) para evaluar un polinomio y se llama **método de Horner**. Este puede expresarse mediante el siguiente pseudocódigo:

```
a=[a0,a1,...,an]      #coeficientes del polinomio
y=a[n]
for(i=1;i<=n;i++){
    y=y*c+a[n-i]
}
```

- Evalúe $3x^2 + x + 1$ en $x = 2$ usando el método de Horner.
- Dé una estimación en notación *big-oh* para el número de operaciones (suma o multiplicación) usadas en el método de Horner.

Ejercicio 9. Indique cuál es el efecto en el tiempo de ejecución de un algoritmo cuando el número de datos se duplica, si se sabe que dicho algoritmo es:

- $O(\log \log n)$
- $O(\log n)$
- $O(n \log n)$
- $O(n^2)$
- $O(2^n)$

Ejercicio 10.

- Describa un algoritmo que encuentre el entero más pequeño en una secuencia finita de números naturales.
- Dé una estimación en notación *big-oh* para el número de comparaciones usadas en el algoritmo.

Modalidad de entrega

- Subir *scan* de su trabajo en PDF a MiU.
- Recuerde escribir su nombre en la hoja.