

1. Para cada uno de los tiempos que toma un algoritmo en terminar, demostrar que cada uno es de la complejidad dada:

a)  $T_A(n) = 2n^3 - 3n^2 + 1$  es  $O(n^3)$

b)  $T_A(n) = n^5 + 4^2 - \sqrt{n} + 1$  es  $O(n^5)$

c)  $T_A(n) = n^2 \log n + 2n^4 + \sqrt{2}n$  es  $O(n^4)$

2. Calcule paso a paso la complejidad del siguiente algoritmo

```
void XXXXXXX(int n) {  
    int x = 0;  
  
    for (int i = 1; i <= n; i *= 5) {  
        int j = 1;  
        for (int j = 1; j <= n; j += 2) {  
            x = x + j;  
        }  
        for (int k = n; k >= 1; k /= 2) {  
            x = x + 1;  
        }  
    }  
}
```

3. Suponga que se tienen los siguientes dos algoritmos para resolver el mismo problema, y suponga que la función **buscar** tiene complejidad  $O(n \log n)$ .

```
boolean Ex1(int[] a, int elem) {  
    int pos = buscar(a, elem);  
    int n = a.length;  
    int x = pos;  
  
    for (int i = 0; i < n; ++i) {  
        x += 2;  
        for (int j = 0; j < n; ++j) {  
            if (a[j] > a[pos]) {  
                x++;  
            }  
        }  
    }  
    return x > elem;  
}
```

```
boolean Ex2(int[] a, int elem) {  
    int n = a.length;  
    int x = 0;  
  
    for (int i = 0; i < n; ++i) {  
        int pos = buscar(a, elem);  
        x += pos + 2;  
        for (int j = 0; j < n; ++j) {  
            if (a[j] > a[pos]) {  
                x++;  
            }  
        }  
    }  
    return x > elem;  
}
```

Indique con cuál de los dos algoritmos se queda para resolver el problema. Justifique muy bien su respuesta.