

# ANÁLISIS Y DISEÑO DE ALGORITMOS

## COMPLEJIDAD TEMPORAL: CÁLCULO ANALÍTICO

### Práctica 3 de laboratorio

Fecha de realización y entrega: del 19 al 24 de febrero de 2015  
(en la sesión de laboratorio que corresponde a cada alumno)

Realiza un análisis de la complejidad temporal de las siguientes funciones del lenguaje C++. En el supuesto de que existan los casos mejor y peor identifica las instancias que pertenecen a cada caso y obtén las correspondientes funciones de complejidad.

Ejercicio 1	Ejercicio 2
<pre>int ejercicio1 (vector &lt; int &gt; &amp; v){      int i,sum=0, n=v.size();      if (n&gt;0){         int j=n;         while (sum&lt;100){             j=j/2;             sum=0;             for (i=j; i&lt;n; i++){                 sum+=v[i];                 if (j==0) sum=100;             }             return j;         }         else return -1;     } }</pre>	<pre>void ejercicio2 (vector &lt; int &gt; &amp; v){     int j, i=1, n=v.size();     if (n&gt;1) do{         int x = v[i];         for (j=i; j &gt; 0 and v[j-1] &gt; x; j--)             v[j]=v[j-1];         v[j]=x;         i++;     } while (i&lt;n); }</pre>

Ejercicio 3
<pre>unsigned ejercicio3 (unsigned n) {     unsigned i=n, k=0;     while (i&gt;0){         unsigned j=i;         do{             j = j * 2;             k = k + 1;         } while (j&lt;=n);         i = i / 2;     }     return k; }</pre>