

# ANÁLISIS Y DISEÑO DE ALGORITMOS

## Complejidad temporal: Cálculo analítico (II)

### Práctica 4 de laboratorio

Fecha de realización y entrega: del 26 de febrero al 3 de marzo 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>float Mochila(vector &lt;float&gt; &amp;v,         vector&lt;unsigned&gt; &amp;p,         unsigned P, int i) {     float S1, S2;     if (i&gt;=0){         if (p[i] &lt;= P)             S1= v[i]+Mochila(v,p,P-p[i],i-1);         else S1= 0;         S2= Mochila(v,p,P,i-1);         return max(S1,S2);     }     return 0; }</pre>	<pre>void abstracto (unsigned n){     if (n&gt;1){         for (unsigned i=1; i&lt;n-1; i++){             for (unsigned j=1; j&lt;=i; j++)                 cout &lt;&lt; "*";             cout &lt;&lt; endl;         }          for (unsigned i=0; i&lt;4; i++)             abstracto(n/2);     } }</pre>
Ejercicio 3	
<pre>bool palind (string &amp; pal, int pri, int ult){      if (pri&gt;=ult) return true;     else{         for (int i=pri; i&lt;ult; i++) cout &lt;&lt; pal[i];         return ( pal[pri]==pal[ult] &amp;&amp;                 palind(pal, pri+1, ult-1) );     } }</pre>	