Obtención de la fórmula maestra.

ALGORITMICA

Francisco Navarro Morales - GRG121

Segundo curso del Grado de Ingeniería Informática
Universidad de Granada
curso 2016-2017

En general, cuando utilizamos la metodología de **divide y vencerás**, nos encontramos con a llamadas recursivas de tamaño n/b para las cuales necesitamos un tiempo $f(n) = d \times n^p \in O(N^P)$ para la división y combinación de los subproblemas.

$$t(n) = at(n/b) + dn^p$$

que con el cambio $n = b^k \rightarrow k = \log_b n$ da lugar a:

$$t(b^k) = at(b^{k-1}) + db^{p^k} \rightarrow t_k - at_{k-1} = db^{p^k} \rightarrow x - a = db^{p^k}$$

Se trata de una ecuación lineal no homogénea. El polinomio característico de la parte homogénea es $x - a$, y de la parte no homogénea obtenemos $x - b^p$

- Si $a = b^p$ entonces tenemos una única raíz de multiplicidad 2: $pc = (x-a)^2$ luego la solución general sería:
  $$t(k) = c_1 b^{p^k} + c_2 k b^{p^k} \rightarrow t(n) = c_1 b^{log_b n^p} + c_2 log_b n \times n^p$$
  es decir:
  $$t(n) = c_1 n^p + c_2 log_b n \times n^p$$
  Como estamos hablando de problemas de eficiencia de algoritmos, es imposible que los coeficientes c_1 o c_2 sean negativos, por lo que podemos afirmar que, en este caso $t(n) \in O(n^p log_b n)$
- Si $a > b^p$ entonces tenemos dos raíces distintas de multiplicidad 1: $pc = (x-a)(x-b^p)$
  $$t(k) = c_1 a^k + c_2 b^{p^k}$$
  donde $a^k > b^{p^k}$ porque $a > b^p$ luego:
  $$t(n) = c_1 a^{log_b n} + c_2 b^{log_b n^p} == c_1 a^{log_b n} + c_2 n^p$$
  Se puede demostrar que $a^{log_b n} = n^{log_b a}$ porque, tomando el log_b de ambas partes nos queda: $log_b n \times log_b a = log_b a \times log_b n$ Luego:
  $$t(n) = c_1 n^{log_b a} + c_2 n^p$$
  y como hemos dicho que $c_1 n^{log_b a} > c_2 n^p$ entonces
  $$t(b) \in O(n^{\log_b a})$$
- Si $a < b^p$ tenemos el mismo resultado que en el caso anterior:
  $$t(n) = c_1 n^{log_b a} + c_2 n^p$$
  Pero en este caso, $c_1 n^{log_b a} < c_2 n^p$ y por lo tanto,
  $$t(b) \in O(n^p)$$