Búsqueda Binaria

Alberto Valderruten

Dept. de Ciencias de la Computación y Tecnologías de la Información
Universidade da Coruña

alberto.valderruten@udc.es

Búsqueda Binaria (1)

- Ejemplo de *algoritmo logarítmico*
- Dados x y un vector *ordenado* a_1, a_2, ... a_n de enteros,

$$\text{devolver: } \begin{cases} i \text{ si } \exists a_i = x \\ \text{“elemento no encontrado”} \end{cases}$$

$\rightarrow$ Comparar x y a_{medio}, con $medio = (i + j) div 2$, siendo $a_i..a_j$ el *espacio de búsqueda*:

1. $x = a_{medio}$: terminar (interrupción)
2. $x > a_{medio}$: seguir buscando en $a_{medio+1}..a_j$
3. $x < a_{medio}$: seguir buscando en $a_i..a_{medio-1}$

- ¿n^o iter? $\leftrightarrow$ evolución del tamaño d del espacio de búsqueda

Invariante: $d = j - i + 1$

$$\text{¿Cómo decrece d? } \begin{cases} i \leftarrow medio + 1 \\ j \leftarrow medio - 1 \end{cases}$$

- *Peor caso*: se alcanza la terminación “normal” del bucle $\equiv i > j$

Búsqueda Binaria (2)

```
    función Búsqueda Binaria (x, a[1..n]) : posición
    {a: vector ordenado de modo no decreciente}

{1}    i := 1 ; j := n ;                        {espacio de búsqueda: i..j}
{2}    mientras i <= j hacer
{3}       medio := (i + j) div 2 ;
{4}       si a[medio] < x entonces
{5}          i := medio + 1
{6}       sino si a[medio] > x entonces
{7}          j := medio - 1
{8}       sino devolver medio                   {se interrumpe el bucle}
       fin mientras;
{9}    devolver "elemento no encontrado"        {fin normal del bucle}
    fin función
```

Búsqueda Binaria (3) - Análisis del peor caso

- Sea $< d, i, j >$ iteración $< d', i', j' >$:
  1. $i \leftarrow medio + 1$:
     $$i' = (i+j)div2 + 1$$
     $$j' = j$$
     $$\begin{aligned} d' = j' - i' + 1 \quad &= j - (i+j)div2 - 1 + 1 \\ &\leq j - (i+j-1)/2 \\ &= (j-i+1)/2 \\ &= d/2 \end{aligned}$$
     $$\boxed{\rightarrow d' \leq d/2}$$
  2. $j \leftarrow medio - 1$:
     $$i' = i$$
     $$j' = (i+j)div2 - 1$$
     $$\begin{aligned} d' = j' - i' + 1 \quad &= (i+j)div2 - i - 1 + 1 \\ &\leq (i+j)/2 - i \\ &< (j-i+1)/2 \\ &= d/2 \end{aligned}$$
     $$\boxed{\rightarrow d' < d/2}$$ *(decrece más rápido)*

Búsqueda Binaria (4) - Análisis del peor caso

- ¿$T(n)$? Sea d_l: d *después* de la l-ésima iteración

  $$\begin{cases} d_0 = n \\ d_l \leq d_{l-1}/2 \ \forall l \geq 1 \end{cases} \quad \text{(inducción)} \rightarrow d_l \leq n/2^l$$

  hasta $d < 1 \rightarrow l = \lceil log_2 n \rceil + 1 = O(logn)$ iteraciones

  Cada iteración es $\Theta(1)$ (reglas) $\Rightarrow T(n) = O(logn)$

- **Razonamiento alternativo**: *pensar en versión recursiva*

  $$T(n) = \begin{cases} 1 & \text{si } n = 0, 1 \\ T(n/2) + 1 & \text{si } n > 1 \end{cases}$$

  Resolver recurrencia (Cf. seminario) $\rightarrow T(n) = \Theta(logn)$

- **Observaciones**:
  - Pensar en versión recursiva puede ser otro recurso útil
  - $\rightarrow$ $\boxed{\text{Algoritmo de reducción}}$
  - $T(n) = \Theta(logn)$

    $\leftrightarrow$ los datos ya están en memoria (Cf. Modelo de Computación)

Búsqueda Binaria (5) - Ejercicios

1. Caracterización del mejor caso (¿cuándo se produce?)
2. Análisis del mejor caso
3. Caracterización del peor caso
4. Demuestre, aplicando las reglas de forma detallada, que cada iteración es $\Theta(1)$
5. Resuelva la recurrencia de la versión recursiva
6. Diseñe el algoritmo recursivo