Algoritmo de Busqueda Binaria Recursiva

Sebastian Ceron, Pedro Salas, Nicolas Oyarzun

Universidad Tecnologica Metropolitana de Chile

26 de Mayo, 2013

¿Que es recursividad?

- Tecnica de Programacion que se llama a si mismo para encontrar la solucion, esta llamada se conoce como recursiva o recurrente
- Existen 2 tipos de Recursividad :

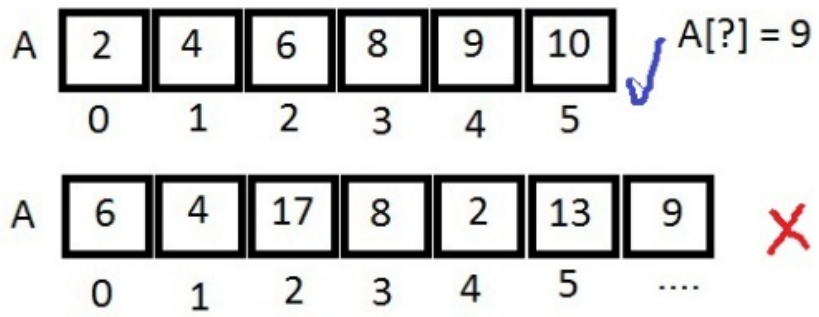
1. Directa : Cuando un subprograma se llama a si mismo una o mas veces
2. Indirecta: Cuando varios subprogramas se utilizan unos con otros

Algoritmo de Busqueda Binaria

- Consiste en reducir el rango de busqueda a la mitad en cada iteracion
- Funcion bajo el concepto “divide y venceremos”
- se utiliza para buscar en lista o vectores

CONDICIONES:

- Elementos ordenados (si no lo estan se puede utilizar metedos de ordenamientos como los SORT)
- Tamaño de la lista o vector
- Elemento a buscar



Busqueda Binaria Recursiva

- La busqueda binaria recursiva funciona exactamente igual que la busqueda binaria iterativa con la diferencia que en vez de utilizar ciclos iterativos se llama a si misma para encontrar la solucion

Ejemplo

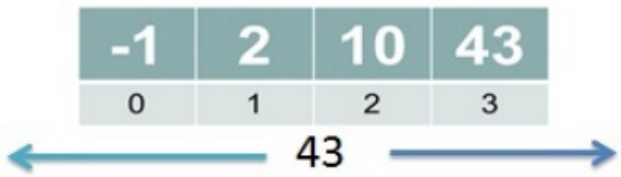
- Supongamos que tenemos la siguiente secuencia :

-1	2	10	43	44	55	60	71	82
0	1	2	3	4	5	6	7	8

- Y buscamos el numero 43
- Ahora escogemos el valor de en medio del arreglo, que es el valor en el índice 4 (el punto medio entre 0 y 8) en este caso el 44
- Comparamos esta mediana con nuestro valor objetivo (43)

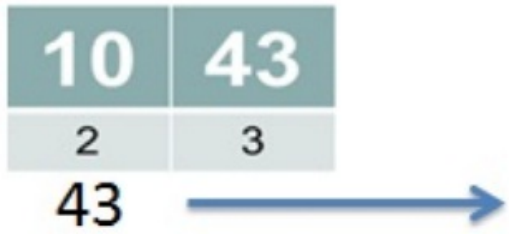
Ejemplo

- Utilizando la propiedad de ordenación del arreglo, podemos concluir que el 43 sólo puede estar en la parte izquierda de la secuencia. (43<44)
- Por lo que descartamos la mitad del espacio de búsqueda que no nos sirve. (Nos quedamos con los elementos 0 a 3).



- El proceso se repite. 43>2 por lo tanto tomamos la mitad de la derecha (elemento en la posición 3).

Ejemplo



- 43>10 por lo tanto se toma la parte derecha

- En 4 operaciones encontramos un elemento en un arreglo ordenado de tamaño 8.

Complejidad

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

$$\begin{aligned} T(n) &= T(n/2) + 1 = \\ &= T(n/2^2) + 2 = \\ &\ldots. \\ &= T(n/2^k) + k \end{aligned}$$

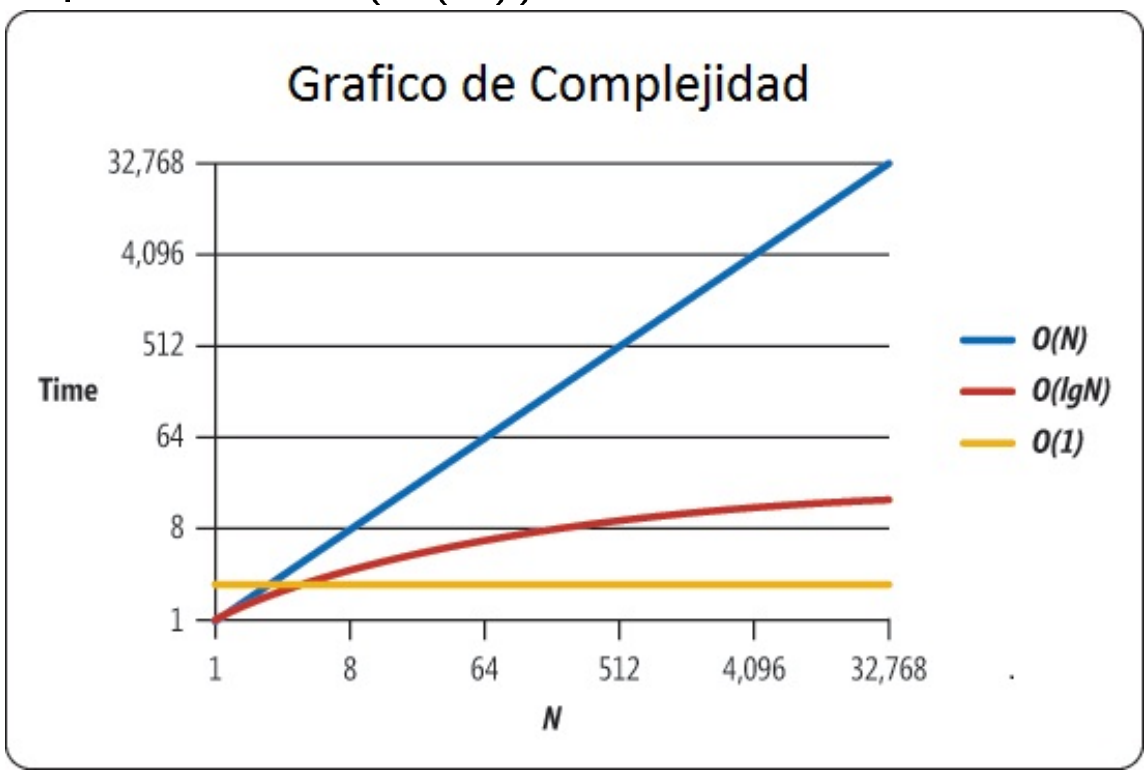
Como la ecuación $n/2^k = 1$ se resuelve para $k = \log_2 n$, tenemos que

$T(n) = T(1) + \log_2 n = 1 + \log_2 n$, y por tanto $T(n)$ es $O(\log n)$.

- El ABB progresivamente va disminuyendo el número de elementos sobre el que realizar la búsqueda a la mitad: n, n/2, n/4, ... Así, tras log (n) divisiones se habrá localizado el elemento o se tendrá la seguridad de que no estaba.

Complejidad

- Como podemos ver en el siguiente grafico, la coplejidad de ABBR tanto en el mejor de los casos (O(1)) como en el caso general (O(LogN)) el tiempo que demora en mucho menor que una busqueda lineal (O(N))



Comparacion

Busqueda Binaria

```
int buscarbi(intvector[],int n, int valor)
 {
intmedio, inicio=0, fin = n-1, encontro = 0;
while (inicio < fin && encontro ==0)
{
medio= (int) (incio+fin) / 2;
if (vector[medio] == valor)
      encontro=1;
else
      if (vector[medio] > valor)
            fin = medio – 1;
      else
            inicio = medio + 1;
}
return encontro
}
```

Busqueda Binaria Recursiva

```
int buscarrec (intvector[],int inicio, int fin,
int valor)
{
intmedio=(int) (inicio+fin) / 2;
if (inicio > fin)
      Return 0;
If (vector[medio]== valor)
      Return 1;
else
      if(vector[medio]>valor)
            Return buscarrec (vector,int
            inicio,medio - 1, valor)
      else
            Return buscarrec(valor,medio
            +1,fin,valor)
}
```

Tabla de Caracteristicas

Búsqueda Binaria (iterativa)	Búsqueda Binaria Recursiva
La función nunca se llama a si misma	Se llama a si misma para realizar la búsqueda
Por lo general utiliza un ciclo while para dividir el Vector o Lista	No utiliza ciclos iterativos
Consume menos recursos que la búsqueda binaria recursiva	Consume mas recursos que la búsqueda binaria iterativa
Tiempo de ejecución Log(n)	Tiempo de ejecución Log(n)
El vector debe estar ordenado	El vector debe estar ordenado
Es necesario conocer el tamaño del vector	Es necesario conocer el tamaño del vector

El siguiente Link direcciona al video explicativo con naipes del algoritmo : http://youtu.be/dXxtj70COJA