Búsqueda Binaria

Madeleine Opazo , Camilo Candia , Jhon Lopez

Análisis de Algoritmos

Introducción

La Búsqueda Binaria utiliza la técnia llamada Divide y Vencerás. Esta consiste en resolver un problema dificil, dividiéndolo en partes más simples, tantas veces como sea necesario, hasta que la resolución de las partes se torna obvia.En particular, este término hace referencia a uno de los más importantes paradigmas de diseño algorítmico.
El método está basado en la resolución recursiva de un problema dividiéndolo en dos o más subproblemas de igual tipo o similar.
En el siguiente trabajo se trata la forma NO recursiva del algoritmo de búsqueda.

Precedentes Históricos

1. Muchos siglos antes de Cristo: Euclides emplea el algoritmo de divide y vencerás con un único subproblema, para computar el máximo común divisor de dos números.
2. 1805: Gauss realiza la descripción del algoritmo divide y vencerás con múltiples subproblemas. Ahora es llamado algoritmo de la rápida transformación de FourierCooley-Tukey (FFT).
3. 1945: John von Neumann inventa el algoritmo de merge-sort, el cual es otro problema antiguo de 2 subdivisiones de divide y vencerás que fue específicamente desarrollado para ordenadores.

Precedentes Históricos

1. 1946: Descripción del algoritmo en ordenadores, publicado en un artículo de John Mauchly.
2. 1960: Algoritmo inventado por A. Karatsuba, que puede multiplicar dos números de n dígitos en $O(n^{log_2 3})$. El algoritmo refutó la conjetura de Andréi Kolmogórov en 1956, de que esa tarea requería (n2) operaciones.

Descripción del Método

Para realizar una búsqueda binaria, se hace lo siguiente:
Se comienza de la suposición de que los elementos o datos numéricos se encuentran previamente ordenados y están ubicados en un arreglo, el cual es llamado Espacio de Búsqueda.
Primero se ubica el dato en el centro del arreglo, es decir, se toma el arreglo original y se divide en dos subarreglos, para luego seleccionar la casilla que se encuentra entre ambos.
Si la casilla que escogimos es igual al dato que buscamos, hemos encontrado la solución y el algoritmo finaliza.

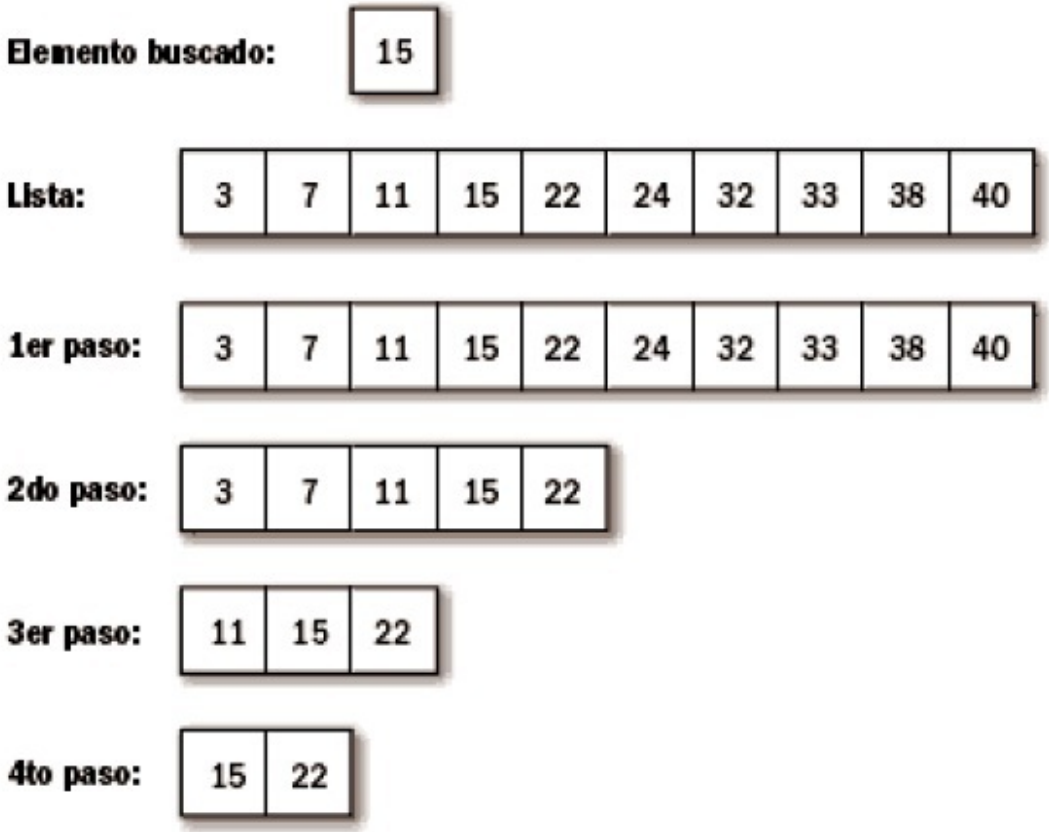
Descripción del Método

Si ocurre que son distintos, revisamos los dos posibles casos:

1. Si el dato en la casilla es más grande que el dato que estamos buscando, los datos de las casillas mayores también lo serán y podemos desecharlas, puesto que es imposible que el dato buscado se encuentre allí.
2. En caso contrario nos deshacemos de las casillas menores.

Cada vez que se realiza lo anterior, el espacio de búsqueda se reduce a la mitad.
Finalmente se aplica el mismo método en las casillas restantes y se repite las veces que sea necesario.

Ejemplo



El algoritmo concluye al encontrar el dato buscado en el primer elemento de la lista, reducida en el último paso a sólo dos elementos.

Codigo C++

```
int busquedaBinaria(int arreglo[], int tamano, int clave)
{
  int Ifinal = tamano-1;
  int Iinicio = 0;
  int Icentro;

  while (Iinicio <= Ifinal)
    {
      Icentro = (Iinicio + Ifinal)/2;
      if (arreglo[Icentro] == clave)
        return Icentro;
      else
      {
        if (clave < arreglo[Icentro])
            Ifinal=Icentro-1;
        else
            Iinicio=Icentro+1;
      }
    }
  return -1;
}
```

Conclusiones

- El algoritmo de búsqueda binaria es muchas veces más rápido, comparado con el método tradicional de búsqueda secuencial (complejidad O(n)), que sería comparar el dato 'objetivo' con cada uno de los elementos del arreglo.
  De esta forma, si se tienen, por ejemplo, 1 millón de datos ordenados en una lista, son necesarias a lo más, 21 comparaciones. Si eso mismo se hiciera empleando la búsqueda secuencial, serian necesarias, en el peor caso, 1 millón de comparaciones.
- Puede que este algoritmo no sea el más rápido a la hora de compararlo con otros que cumplen la misma función, pero es realmente confiable, en parte, debido a su sencillez de código.

Conclusiones

- La desventaja que representa este método es lo inutil que se vuelve al usarlo si los datos están desordenados, para lo cual es necesario emplear, previamente, algun metodo de ordenamiento. Esto hace que se pierda la eficacia, pues se emplea tiempo valioso en el ordenamiento de los elementos.