

Código Fuente

```
#include <stdio.h>
main(){
    int a[]={10,8,7,2,1,3,5,4,6,9},i,aux,der,izq,centro,j;
    for (i=1;i<10;i++){
        aux=a[i]; //Asigna el valor a insertar
        izq=0; //Siempre inicia con 0 (pos 0)
        der=i-1; //lim derecho (ya ordenados) al iniciar iteración
        while(izq<=der){ /*Mientras no lleguen a cruzarse
            completamente estas posiciones continua*/
            centro=(izq+der)/2; //Calculamos centro
            if (aux<=a[centro]) /*Decremento en los
                limites*/
                der=centro-1;
            else
                izq=centro+1;
        }
        j=i-1; /*Cantidad de números a recorrer*/
        while(j>=izq){ /*Recorremos de derecha a izq*/
            a[j+1]=a[j]; //Movemos al siguiente pos
            j--;
        }
        a[izq]=aux; /*Al final insertamos el valor en su posición
            correspondiente*/
    }
    for (i=0;i<10;i++) /*Imprimimos el número de elementos*/
        printf("%d.",a[i]);
    putchar("\n");
    getchar();
}
/*Fin del método de inserción binaria*/
```

Más información

Correo electrónico

llopez17@uabc.edu.mx

Página web

<https://sites.google.com/uabc.edu.mx/lpez-valencia-luis/p%C3%A1gina-principal>

Referencias

María Adriana Corona Nakamura, María de los Ángeles Ancona Valdez. (2011). DISEÑO DE ALGORITMOS Y SU CODIFICACIÓN EN LENGUAJE C. México, D.F.: Mc Graw Hill.

Luis Joyanes Aguilar, Luis Zahonero Martínez. (2005). Programación en C: metodología, algoritmos y estructura de datos. España: McGraw Hill.

Luis Joyanes Aguilar, Luis Rodríguez Baena, Matilde Fernández Azuela. (1996). Fundamentos de Programación. España: Mc Graw Hill.

<https://www.fing.edu.uy/tecnoinf/mvd/cursos/eda/material/teo/FDA-teorico4.pdf>

Método de inserción Binaria



Materia: *Algoritmos y Estructura de Datos*

Alumno: *López Valencia Luis Angel*

Matricula: *1262057*

Profesora: *Palacios Guerrero Alma Leticia*

¿Cómo funciona?

El algoritmo de inserción binaria es solo un método derivado del algoritmo de inserción secuencial, el cual su objetivo principal es insertar el valor de un dato en el lugar que le corresponde. Para hacer esto es necesario ir comparando cada elemento a partir de la posición 1 del arreglo, esto hace que para un tamaño grande de elementos sea el tiempo ejecución más largo.

La mejora del método de inserción binaria llega cuando en la búsqueda del elemento se basa en el algoritmo de búsqueda binaria para reducir el número de comparaciones con respecto al número de elementos en el arreglo, de tal forma que para n número de elementos en un arreglo este pueda realizar en tiempo de ejecución $n/2$ comparaciones por cada elemento a insertar y esto en segundos ahorra el tiempo.

¿Dónde se originó?

Al igual que el algoritmo de inserción secuencial no es un método que solo se haya logrado descubrir en un solo lugar, cualquier persona con el tiempo necesario podría llegar a la misma conclusión al realizar este método, pero en algoritmo de inserción binaria cuenta con una licencia de código abierto MIT/X11 y se puede deducir que el algoritmo se originó en el Instituto de Tecnología de Massachusetts (MIT).

Elementos necesarios

Dentro del algoritmo cuando se quiere implementar en un código, los elementos más importantes que necesita tener para funcionar son:

- Array[n]: El arreglo puede ser de n elementos que ya están definidos en sus posiciones y se quiera insertar valores de forma binaria.
- Centro: Esta variable nos ayudará a comparar el centro obtenido con el valor del dato que queramos insertar y tomar decisiones que cambien los límites inferiores y superiores,
- Variable de control <aux>: En esta variable colocaremos el valor que nosotros queramos insertar en el arreglo.
- Variable de control <izq>: En toda iteración que vaya avanzado 'i' siempre iniciará el lado izq con el valor de 0 (definiendo el lim inferior)
- Variable de control <der>: En toda iteración que vaya avanzado 'i' siempre se colocará el valor en el que vaya 'i'-1 para representar hasta qué posición está ordenado correctamente (definiendo el límite superior).

Estos elementos son los principales para poder encontrar la posición en la que se insertará el valor auxiliar, claro está que dentro del método el lado derecho decrementará o el lado izquierdo incrementada por cada búsqueda

binaria que se haga, esto dependiendo del valor con el que se vaya cambiando el centro = $(izq + der) / 2$

Situaciones

El loop comienza con $der - izq = n - 1$ y termina con $izq > der$ (en el peor caso).

A lo largo del loop el valor $der - izq$ se divide al menos a la mitad de su valor anterior luego la cantidad de iteraciones será de la forma $\log_2 n$.

En el peor de los casos:

En el peor de los casos, el arreglo estará inversamente ordenado . $F(n) = O(n^2)$.

En cualquier otro caso: El número de comparaciones y el número de asignaciones es de orden cuadrático.

$$F(n) = \theta(n^2)$$

En el mejor de los casos: El arreglo estará inicialmente ordenado.

$$F(n) = \Omega(n \log n)$$