**1.Define el problema**

Se necesita realizar el ordenamiento de un arreglo de números que el usuario ingrese al sistema, estos números serán ingresados de forma aleatoria y el programa tendrá que realizar el ordenamiento correspondiente de los mismos.

Por ejemplo, el usuario indica una secuencia de **ENTRADAS** de n números dentro del código :

{0, 9, 1, 7, 3, 8, 2, 6, 4}

Una vez que el código se ejecuta a la **SALIDA** mostrara el arreglo ordenado o una permutación (reordenamiento de los números) de tal forma que por lo tanto el arreglo ya ordenado seria.

0, 1, 2, 3, 4, 6, 7, 8, 9

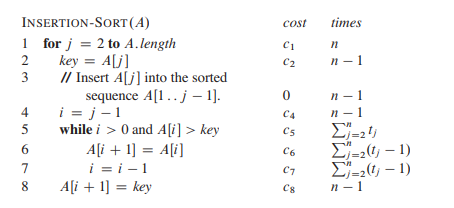
**2.Presenta una propuesta de solución**

El ordenamiento de un arreglo de n números puede compararse a la misma manera que se clasifica una mano de cartas. Comenzamos con una mano izquierda vacía y las cartas boca abajo en la mesa. Luego retiramos una carta a la vez de la mesa y la insertamos en la posición correcta en la mano izquierda. Para encontrar la posición correcta de una carta, la comparamos con cada una de las cartas que ya están en la mano.

Dicho lo anterior el funcionamiento del código, debe de comparar cada número e irlo clasificando para poder realizar el ordenamiento, en la imagen se muestra un pequeño fragmento de Inserción, donde el autor hace un conteo de costos y de tiempo que tarda en ejecutarse dicho código.

Cabe mencionar que los tiempos durante la ejecución del programa son en microsegundos y cada instrucción tiene una duración de un ciclo de reloj de I microsegundo, el autor realiza una sumatoria de todas instrucciones del programa incluyendo las cíclicas hasta llegar a una sumatoria lineal.

Los procesos que tienen que ver durante la ejecución de cualquier programa son la, frecuencia de reloj, el procesador y la cantidad de memoria que.



**EXPLICACIÓN DEL CODIGO SINSERCIÓN DEL AUTOR.**

1.- Inicia con una instrucción de control FOR de j = 2 a n veces, las cuales son el número de veces del ciclo

2.- Utiliza variable key que almacenará la longitud del arreglo.

3.- Indica un comentario dentro del código.

4.- Realiza un decremento de -1 a J y se almacenara en i

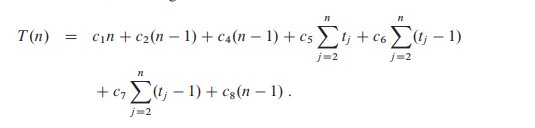
5.- Con la sentencia de control whiles realiza la comparación de i y el arreglo de A[]

6.-Ingresa en A el arreglo de i + 1

7.- realiza el decremento de la variable i -1

8.- Muestra el arreglo de A[] y lo iguala a la variable key

Como se puede observar durante todo esa parte de código las instrucciones generan un costo y un tiempo de ejecución, el autor indica esa sumatoria de la siguiente manera:



Al realizar las sumatoria anterior se puede determinar que la sumatoria total es una suma lineal.



.

**3.Diseña un algoritmo**

Este algoritmo es el propuesto por el autor para poder desarrollar el ordenamiento del arreglo.



**4.Convierte el algoritmo en un programa**

Código donde se ingresa el arreglo manualmente y el código lo ordena automáticamente.

**package** com.tcca;

**public** **class** Insertar {

**public** **static** **void** main(String[] args) {

**int** A[]= {7, 1, 9, 2, 8, 6, 0};

*mostrarArreglo*(A);

*OrdenarIngreso*(A);

*mostrarArreglo*(A);

}

**public** **static** **void** OrdenarIngreso (**int** a []) {

**int** clave, i, N=a.length;

**for** (**int** j=1; j<N; j++) {

clave = a[j];

i = j - 1;

**while** (i>-1 && a[i] > clave) {

a[i+1] = a[i];

i = i-1;

}

a[i+1] = clave;

}

}

**public** **static** **void** mostrarArreglo(**int** a[]) {

**for** (**int** i=1; i<=a.length; i++ ) {

System.***out***.print(a[i-1]+" ");

}

System.***out***.print("\n");

}

}

**5.Verifica la solución**

El código al ejecutarse muestra el arreglo que se ingresó al principio del código y una vez que se realiza la ejecución del programa, muestra el arreglo ya ordenado en la parte de abajo, como se muestra en la figura 1.

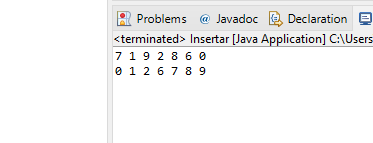


Figura1