

Ejercicio 5: Dependencia de la implementación

Considere esta otra implementación del algoritmo de la burbuja:

En ella se ha introducido una variable que permite saber si, en una de las iteraciones del bucle externo no se ha modificado el vector. Si esto ocurre significa que ya está ordenado y no hay que continuar.

Considere ahora la situación del mejor caso posible en la que el vector de entrada ya está ordenado. ¿Cuál sería la eficiencia teórica en ese mejor caso? Muestre la gráfica con la eficiencia empírica y compruebe si se ajusta a la previsión.

- 1) Creamos el fichero 'mejorcasonuevo.cpp' en el que incluimos tanto el programa completo como el nuevo algoritmo de ordenación de burbuja.

- 2) Generamos el ejecutable mediante:

```
g++ mejor_caso_nuevo.cpp
```

- 3) Modificamos los campos del archivo 'mejor_caso_nuevo.csh' con los datos que nos han pedido.

- 4) Para que el resultado de todas las ejecuciones se guarden en el archivo 'mejor_caso_nuevo.dat' ejecutamos lo siguiente:

```
./mejor_caso_nuevo.csh
```

- 5) Calculamos ahora la eficiencia teórica del algoritmo:

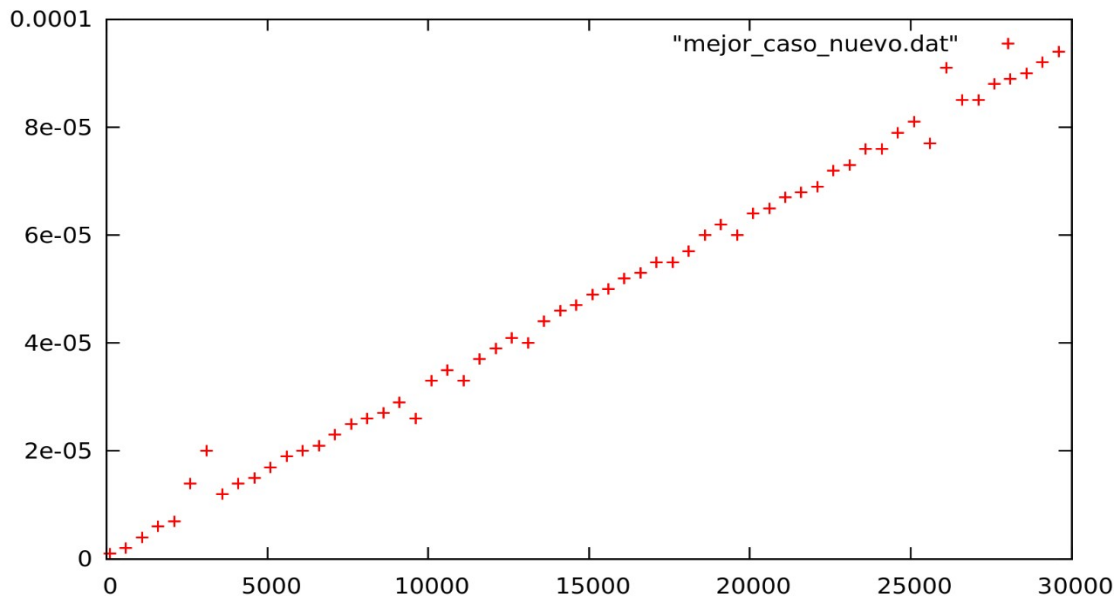
```
void ordenar(int *v, int n) {
    bool cambio=true;
    for (int i=0; i<n-1 && cambio; i++){
        cambio=false;
        for (int j=0; j<n-i-1; j++){
            if (v[j]>v[j+1]){
                cambio=true;
                int aux = v[j];
                v[j] = v[j+1];
                v[j+1] = aux;
            }
        }
    }
}
```

/* Empezamos analizando el if: Como todas sus líneas de código son asignaciones y acceso al vector, por la regla de la suma $O(\max(1,1,1,1))=O(1)$. El segundo for realiza $n-i-1$ iteraciones por lo que tiene una eficiencia de $O(n)$. Lo mismo ocurre con el primer for que realiza $n-1$ iteraciones. Cabe destacar que el hecho de declarar el booleano 'cambio', no afecta a la eficiencia del algoritmo cuando tiende a infinito. Por la regla del producto, cabe destacar que nos encontramos ante una eficiencia cuadrática: $O(n*n*1)=O(n^2)$

- 6) Con la ayuda de gnuplot, una vez obtenido el archivo .dat, representamos su eficiencia empírica:

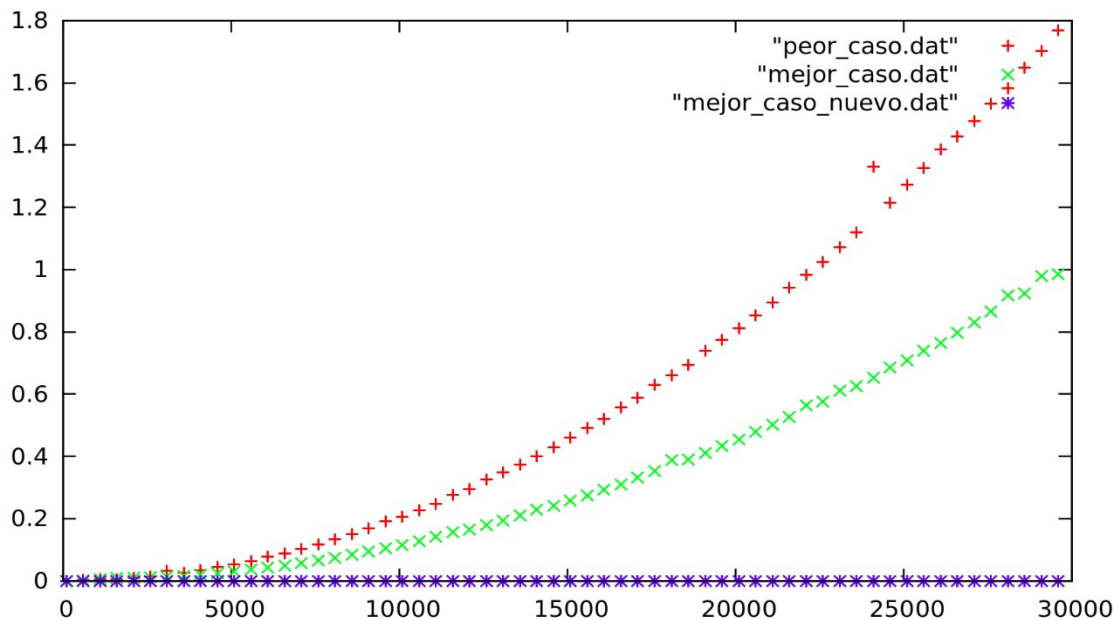
```
>gnuplot  
>plot "mejor_caso_nuevo.dat"
```

La gráfica resultante es:



Como podemos observar, dicho algoritmo realiza los cálculos muy rápido, en el orden de 30000 datos en apenas 0.0001 segundos.

En comparación con las del ejercicio anterior, obtenemos:



Como muestra la gráfica, apenas se aprecia el crecimiento en el mejor_caso_nuevo.dat, no como en el mejor_caso.dat que realiza las mismas operaciones en 1 segundo

Características del ordenador:

Fabricante	Lenovo
Procesador	Intel Core i7 4710HQ
Sistema Operativo	Ubuntu
Versión SO	14.04
RAM	8192MB
CPU	64 bits