EJERCICIO 5 : Dependencia de la implementación

Autor : Jesús Ruiz Castellano, 76439001-L

1.- Código fuente : burbujaModMej.cpp

```cpp
// Algoritmo ordenacion por burbuja modificado
void ordenarModificacion(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;
                        }
        }
}


void sintaxis()
{
  cerr << "Sintaxis:" << endl;
  cerr << "  TAM: Tamaño del vector (>0)" << endl;
  cerr << "  VMAX: Valor máximo (>0)" << endl;
  cerr << "Se genera un vector de tamaño TAM con elementos aleatorios en [0,VMAX[" << endl;
  exit(EXIT_FAILURE);
}
int main(int argc, char * argv[])
{
  // Lectura de parámetros
  if (argc!=3)
    sintaxis();
  int tam=atoi(argv[1]);     // Tamaño del vector
  int vmax=atoi(argv[2]);    // Valor máximo
  if (tam<=0 || vmax<=0)
    sintaxis();

  // Generación del vector
  int *v=new int[tam];       // Reserva de memoria


  // Rellenar el vector con datos ordenados, para el mejor caso
  for (int i=0; i<tam; i++)  // Recorrer vector
    v[i] = i;                // Le pasamos un vector ordenado de menor a mayor


  clock_t tini;    // Anotamos el tiempo de inicio
  tini=clock();

  for (int i = 0 ; i < 1000 ; i++)
      ordenarModificacion(v,tam);          // v esta ordenado, que es el mejor caso

  clock_t tfin;    // Anotamos el tiempo de finalización
  tfin=clock();

  // Mostramos resultados reduciendo el error en 1000
  cout << tam << "\t" << ((tfin-tini)/(double)CLOCKS_PER_SEC)/1000.0 << endl;

  delete [] v;     // Liberamos memoria dinámica
}
```

EJERCICIO 5 : Dependencia de la implementación

2. Hardware usado:

2.1- CPU

vendor_id : GenuineIntel
model name : Intel(R) Core(TM) i3 CPU M 330 @ 2.13GHz
cpu MHz : 933.000

2.2- Velocidad de Reloj

Versión : hwclock de util-linux 2.20.1
mie 12 oct 2016 00:50:11 CEST -0.563198 segundos

2.3- Memoria RAM

MemTotal : 3907668 kB
SwapTotal : 4049916 kB

3.- Sistema Operativo

Ubuntu 14.04.3 LTS
Arquitectura : x86_64 (64 bits)

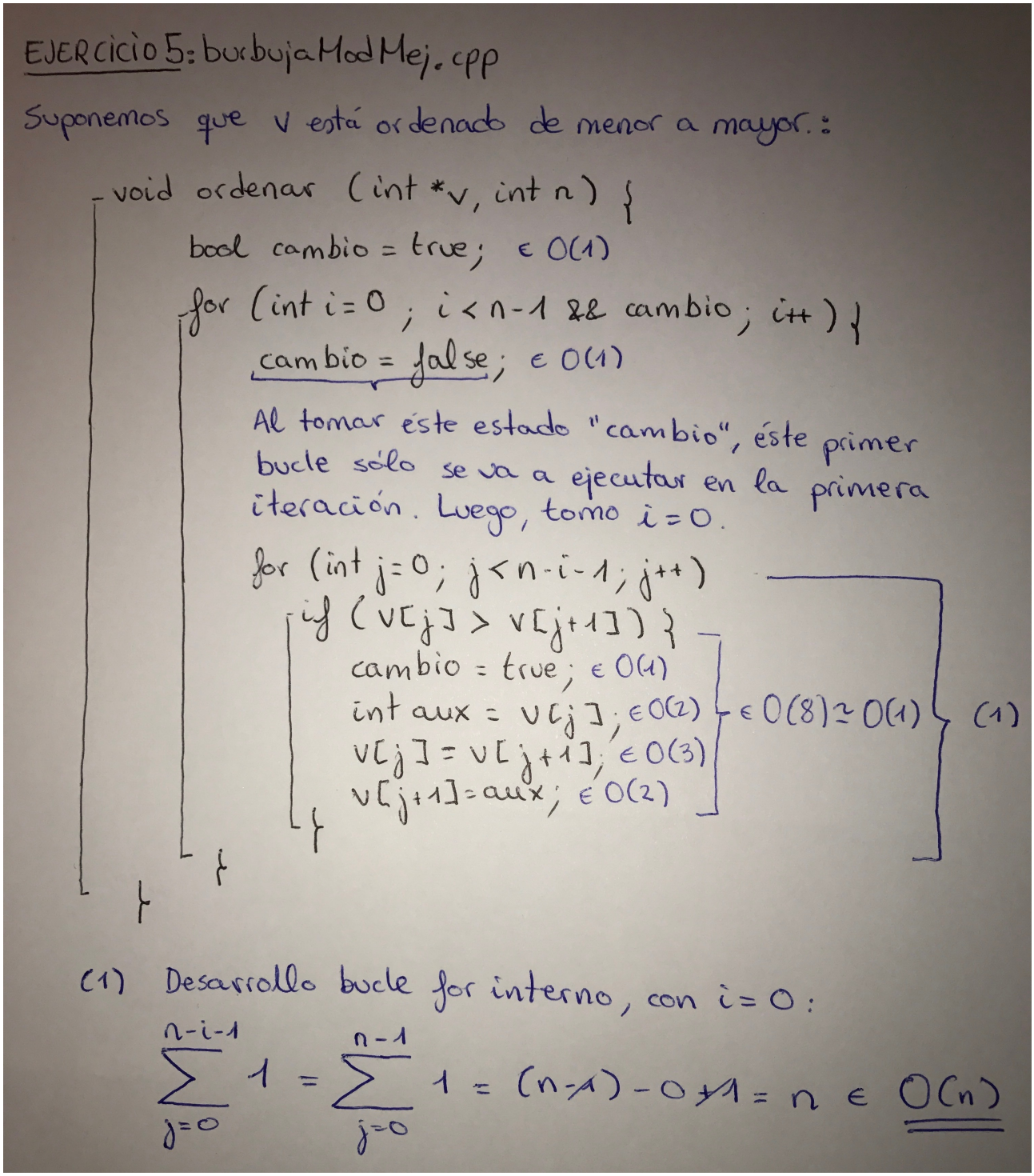
4.- Compilador usado y opciones de compilación

gcc - GNU project C and C++ compiler
Opción de compilación : g++ -o <nombre_ejecutable> <ejecutable.cpp>
g++ -o burbujaModMej burbujaModMej.cpp

EJERCICIO 5 : Dependencia de la implementación

5.- Desarrollo completo del cálculo de la eficiencia teórica

MEJOR CASO



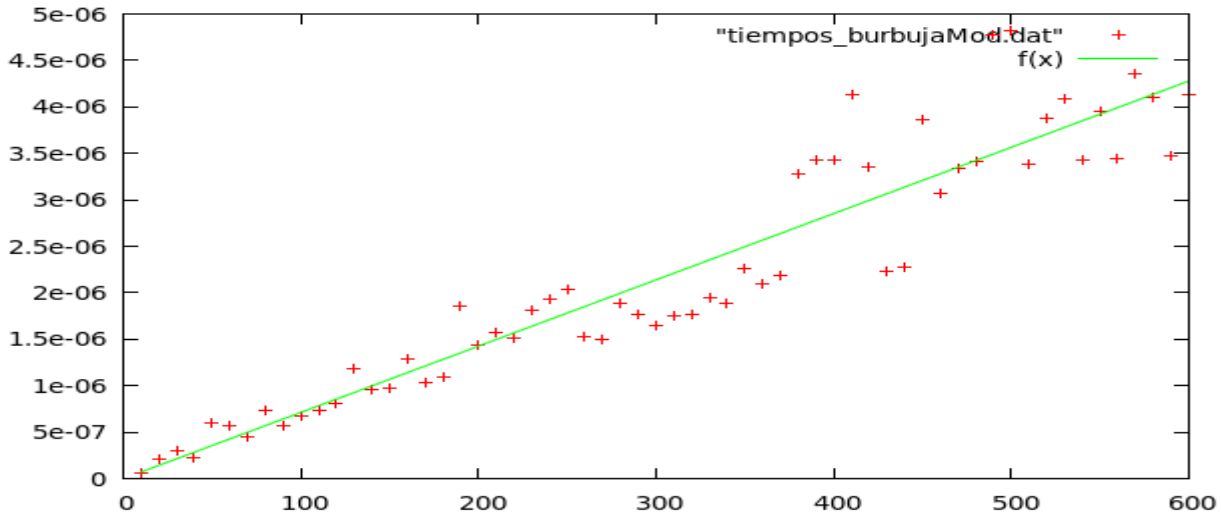
EJERCICIO 5 : Dependencia de la implementación

6.- Parámetros usados para el cálculo de la eficiencia empírica y gráfica

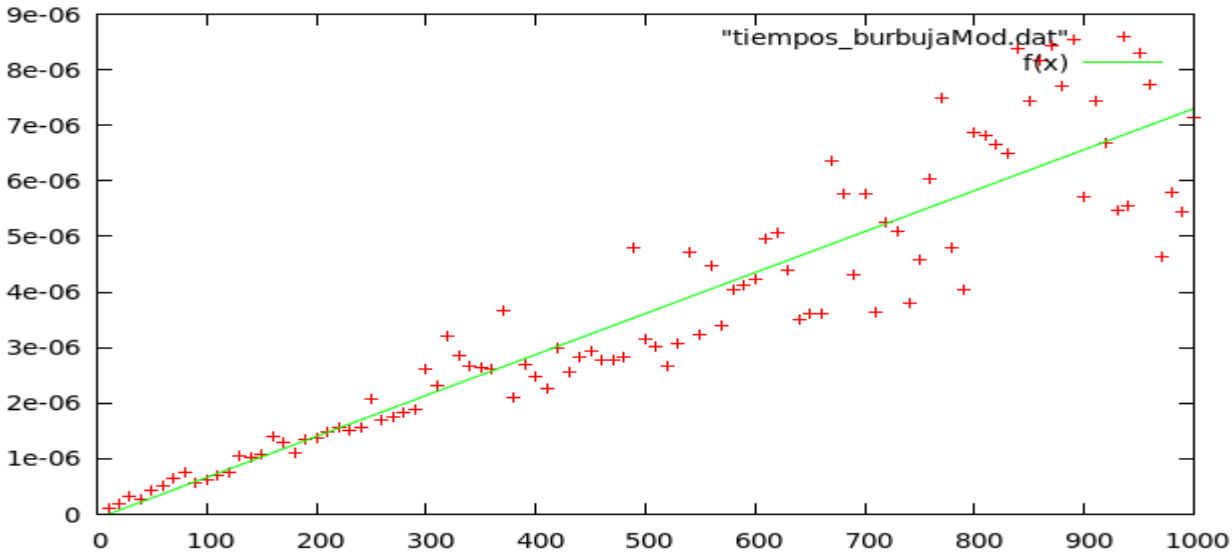
MEJOR CASO

Para ésta parte he ejecutado el programa con los siguientes valores para tamaño máximo del vector : 600, 1000, 3000, 6000, 10000 y 20000.

6.1- Gráfica de tiempos para la ejecución del programa con tamaño de vector = 600

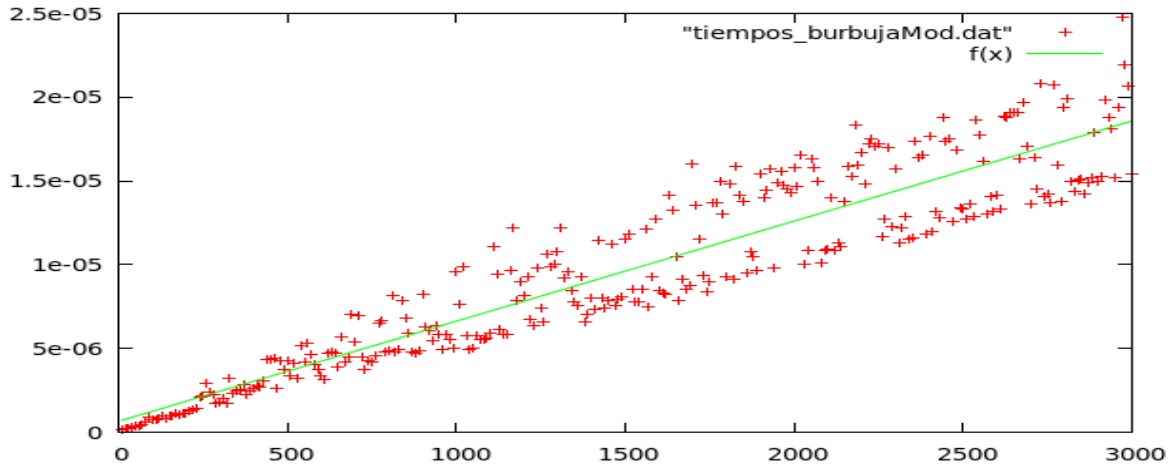


6.2- Gráfica de tiempos para la ejecución del programa con tamaño de vector = 1000

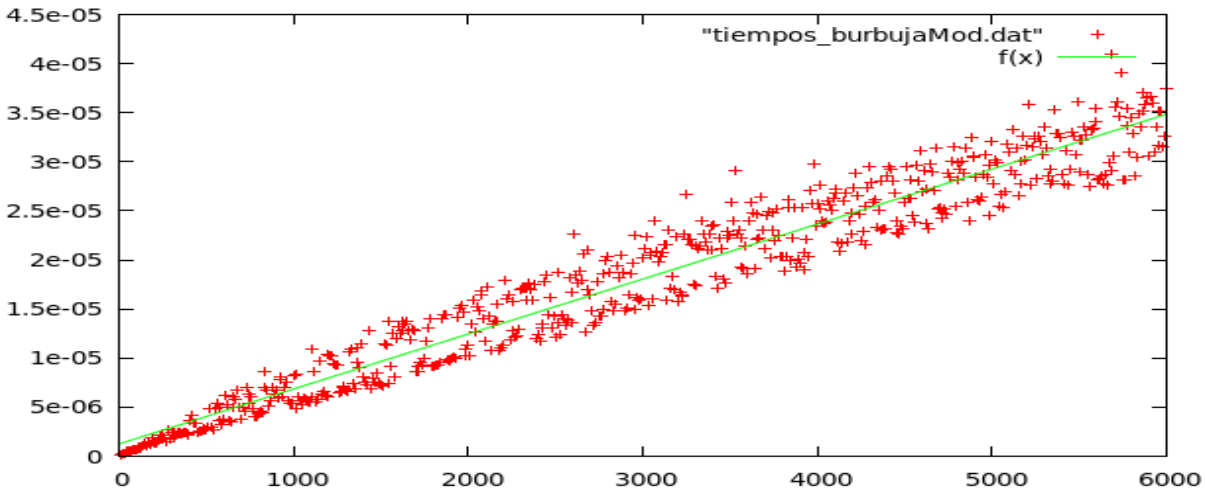


EJERCICIO 5 : Dependencia de la implementación

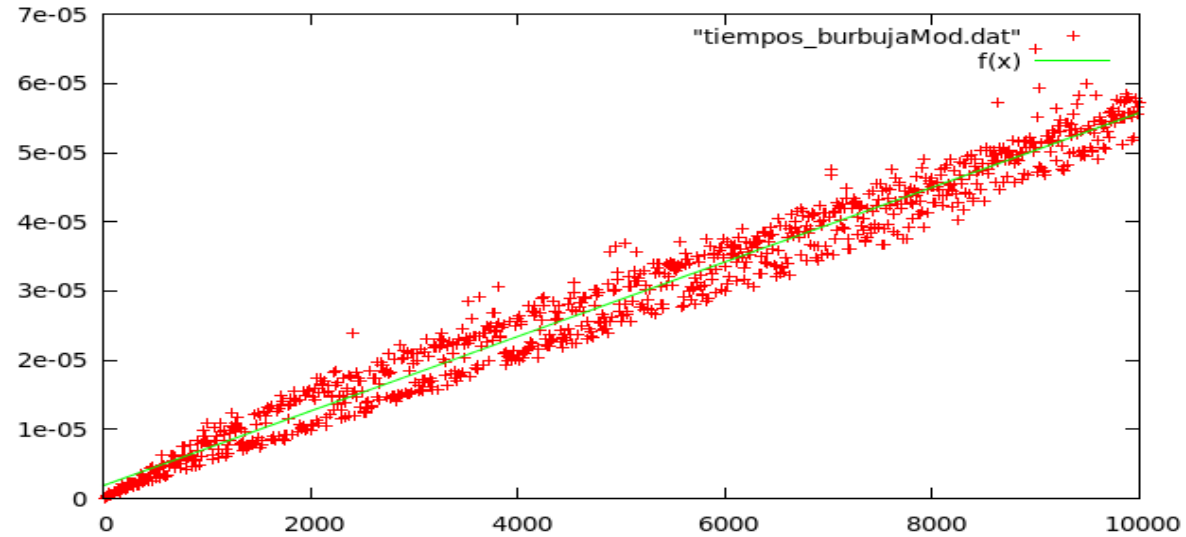
6.3- Gráfica de tiempos para la ejecución del programa con tamaño de vector = 3000



6.4- Gráfica de tiempos para la ejecución del programa con tamaño de vector = 6000



6.5- Gráfica de tiempos para la ejecución del programa con tamaño de vector = 10000



EJERCICIO 5 : Dependencia de la implementación

6.6- Gráfica de tiempos para la ejecución del programa con tamaño de vector = 20000