

David Infante Casas
Pablo Rodríguez Vargas

Algorítmica – Práctica 1 – Ejercicio 4 - 2ºD

4.- Mergesort:

- Código Fuente:

```
static void mergesort_lims(int T[], int inicial, int final) {
    if (final - inicial < UMBRAL_MS) {
        insercion_lims(T, inicial, final);
    } else {
        int k = (final - inicial)/2;

        int * U = new int [k - inicial + 1];
        assert(U);
        int l, l2;
        for (l = 0, l2 = inicial; l < k; l++, l2++)
            U[l] = T[l2];
        U[l] = INT_MAX;

        int * V = new int [final - k + 1];
        assert(V);
        for (l = 0, l2 = k; l < final - k; l++, l2++)
            V[l] = T[l2];
        V[l] = INT_MAX;

        mergesort_lims(U, 0, k);
        mergesort_lims(V, 0, final - k);
        fusion(T, inicial, final, U, V);
        delete [] U;
        delete [] V;
    }
};
```

- Hardware:

Procesador: Intel® Core™ i5-3350P CPU @ 3.10GHz × 4
Memoria RAM: 8GB

- Sistema Operativo:

Ubuntu 14.04

- Compilador Utilizado:

g++ -std=c++11

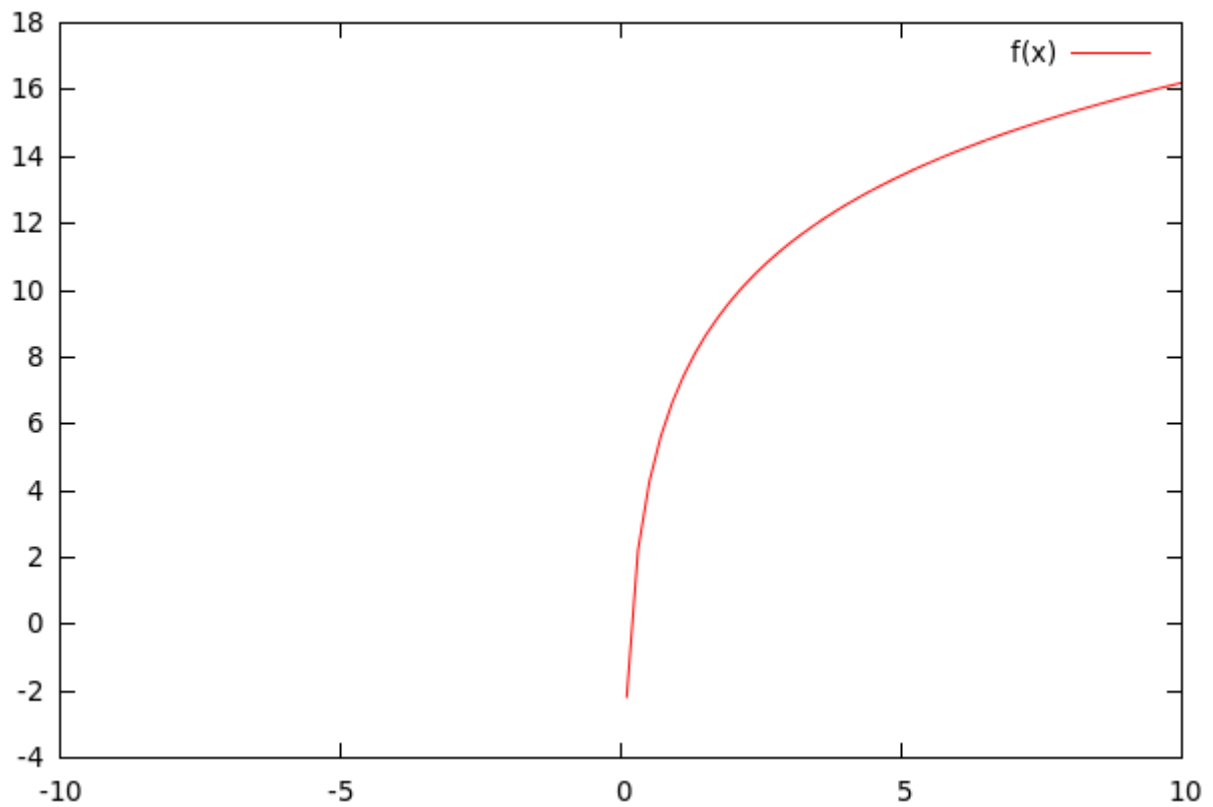
a) Caso Mejor:

- Eficiencia Teórica:

$$T_m(n) = 4\log(n) + 7$$

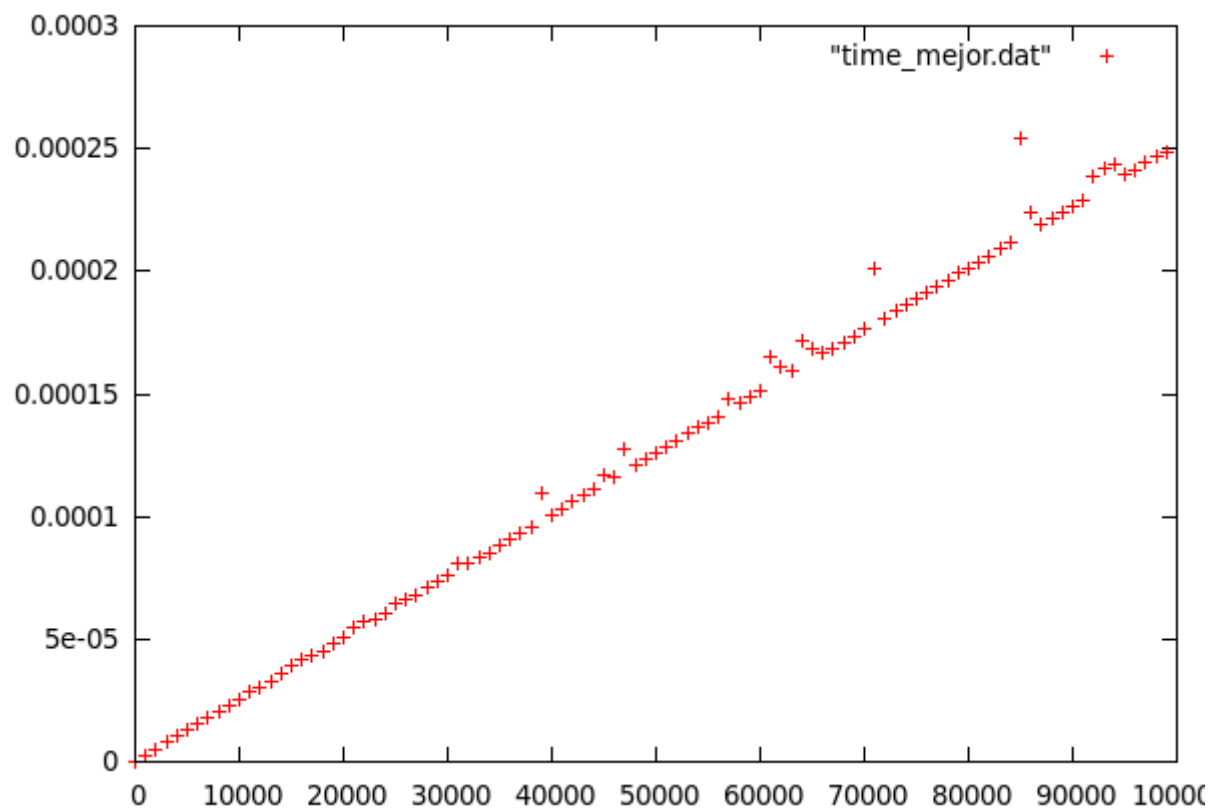
$$f(x) = 4*\log(x)+7$$

plot f(x)



- Eficiencia empírica:

plot "time_mejor.dat"

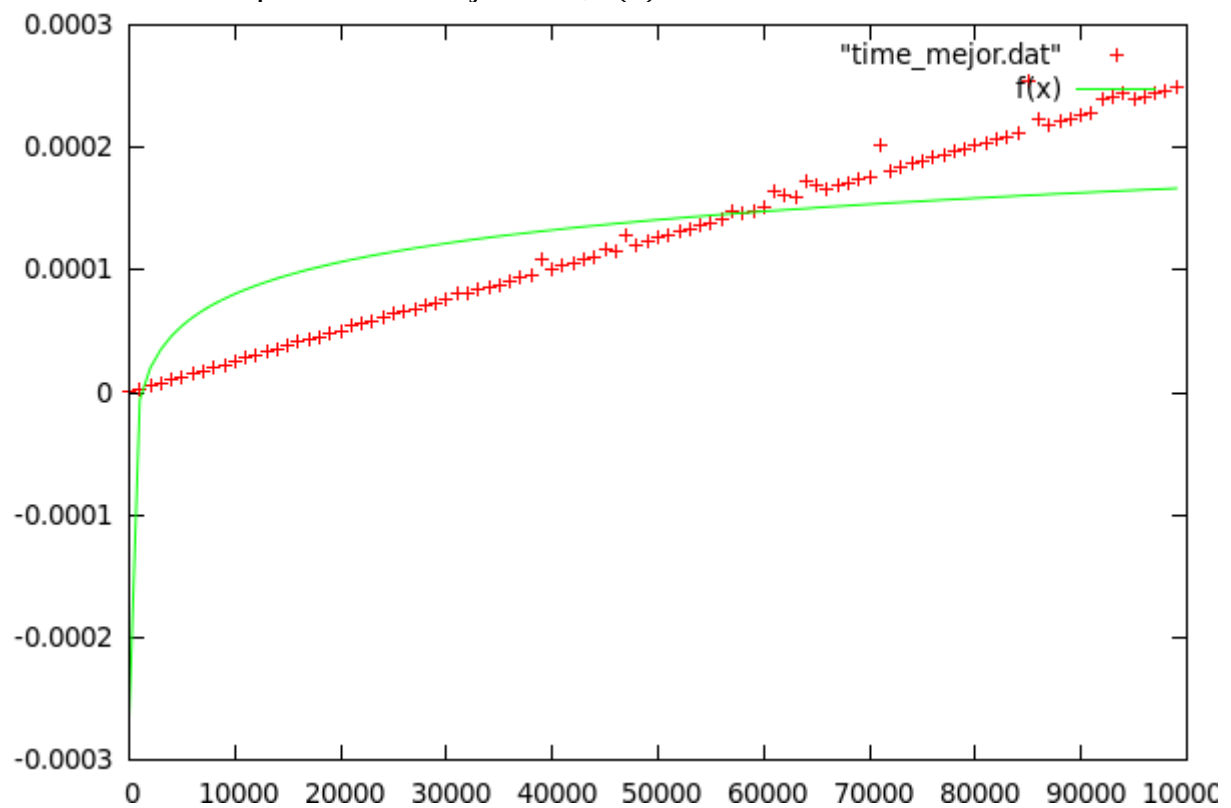


- Ajuste:

$$f(x) = a \cdot \log(x) + b$$

fit $f(x)$ "time_mejor.dat" via a, b

plot "time_mejor.dat", $f(x)$



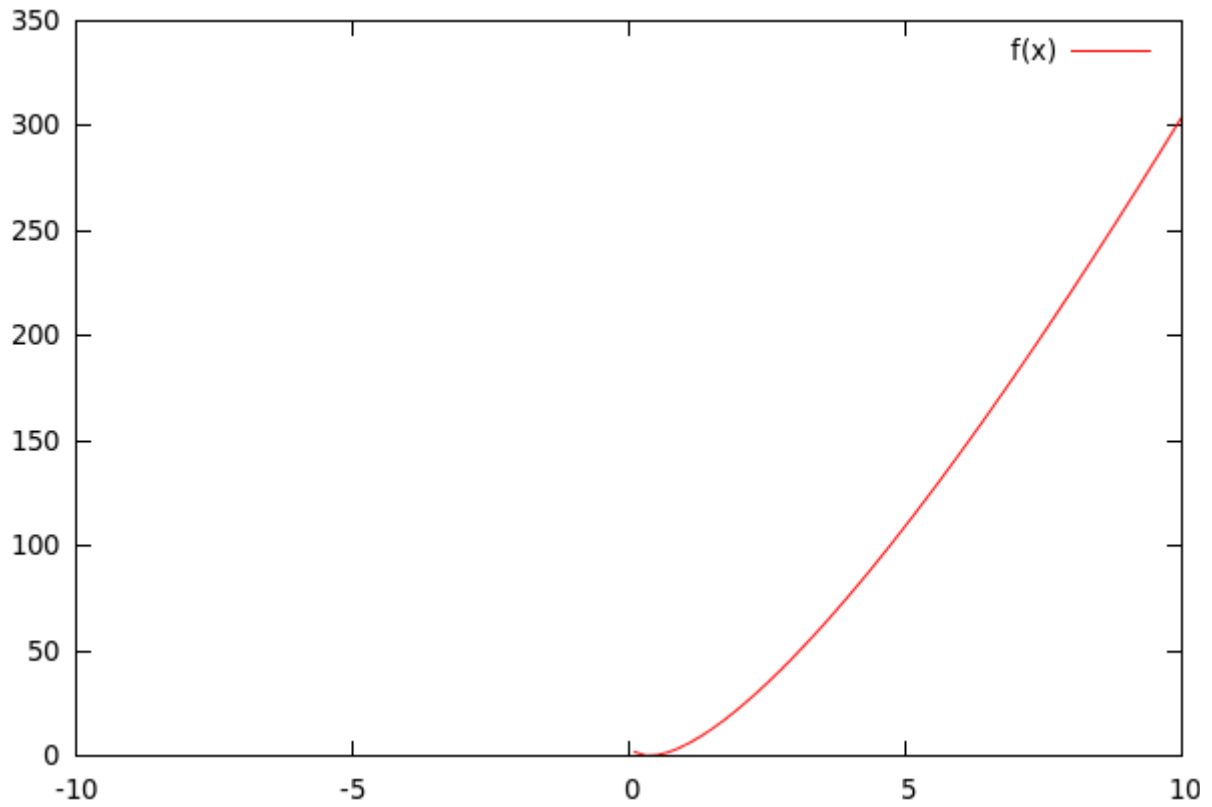
b) Caso Peor:

- Eficiencia Teórica:

$$T_p(n) = 13n\log(n) + 5$$

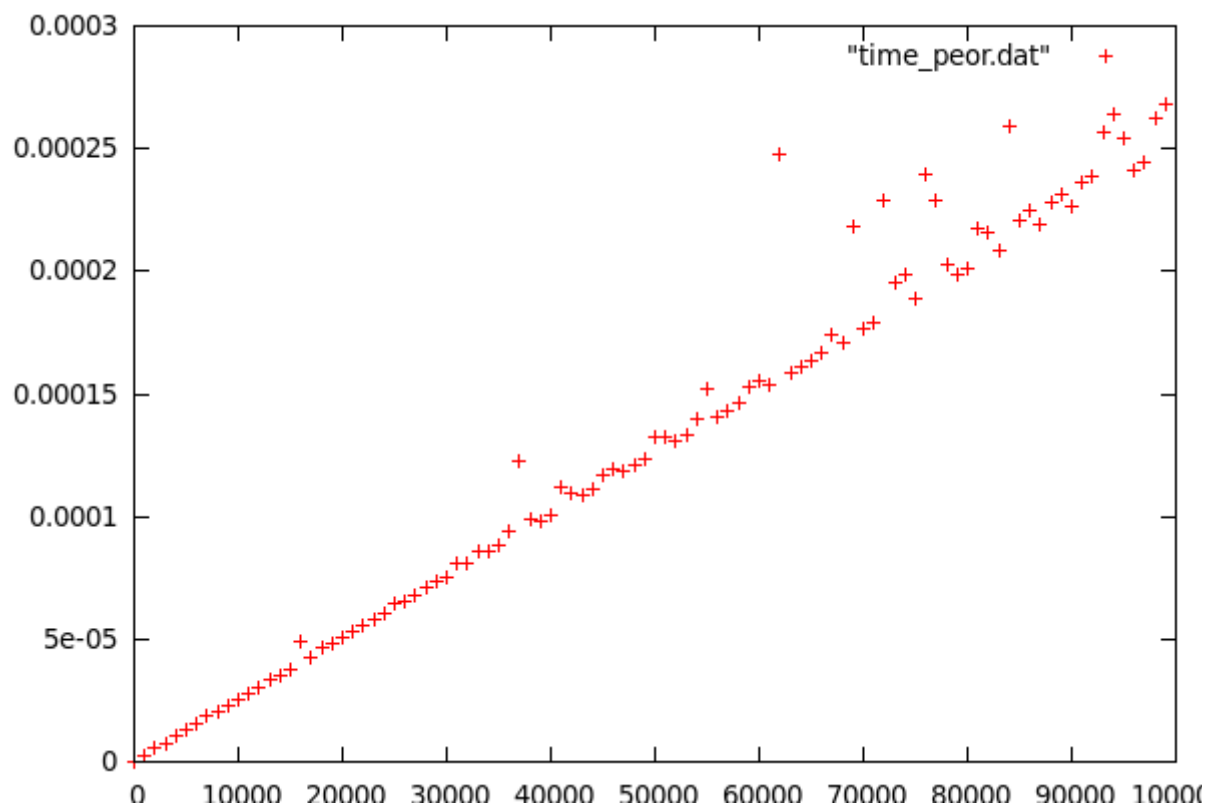
$$f(x) = 13 \cdot x \cdot \log(x) + 5$$

plot f(x)



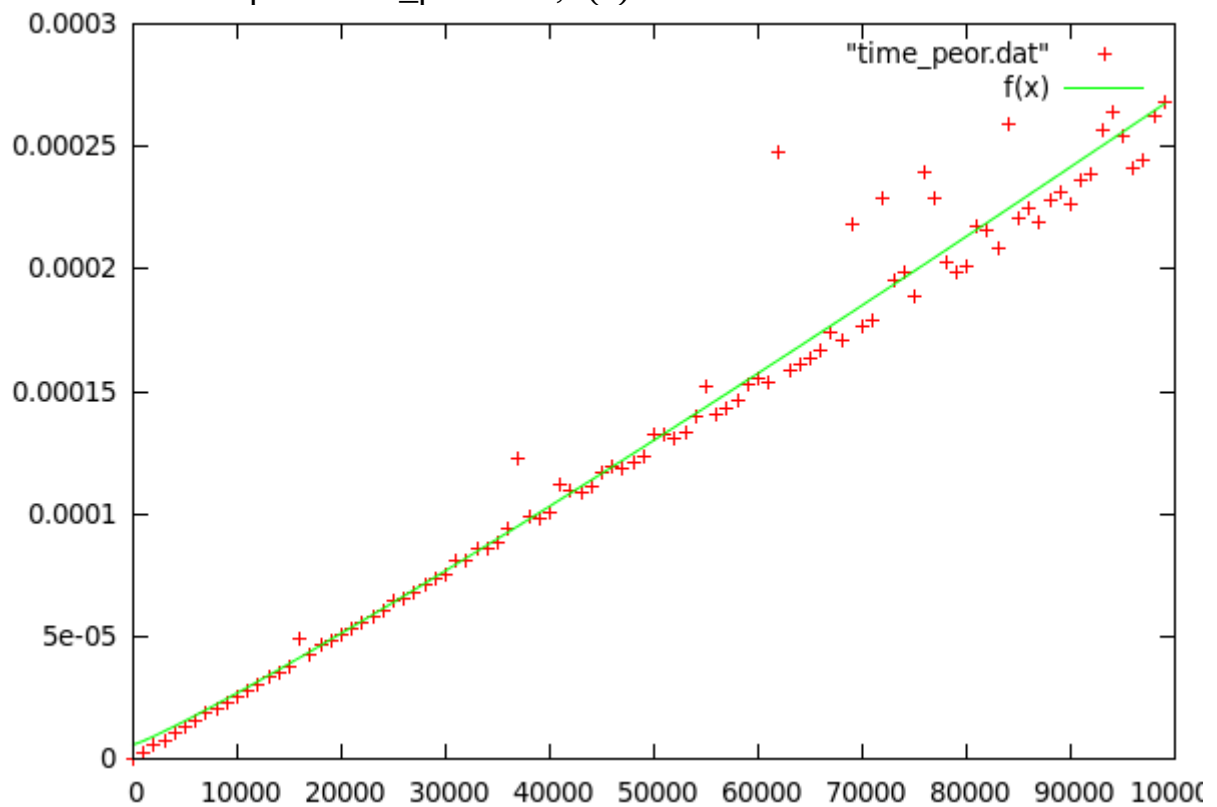
- Eficiencia empírica:

plot "time_peor.dat"



- Ajuste:

$f(x) = a \cdot x \cdot \log(x) + b$
 fit $f(x)$ "time_peor.dat" via a, b
 plot "time_peor.dat", $f(x)$



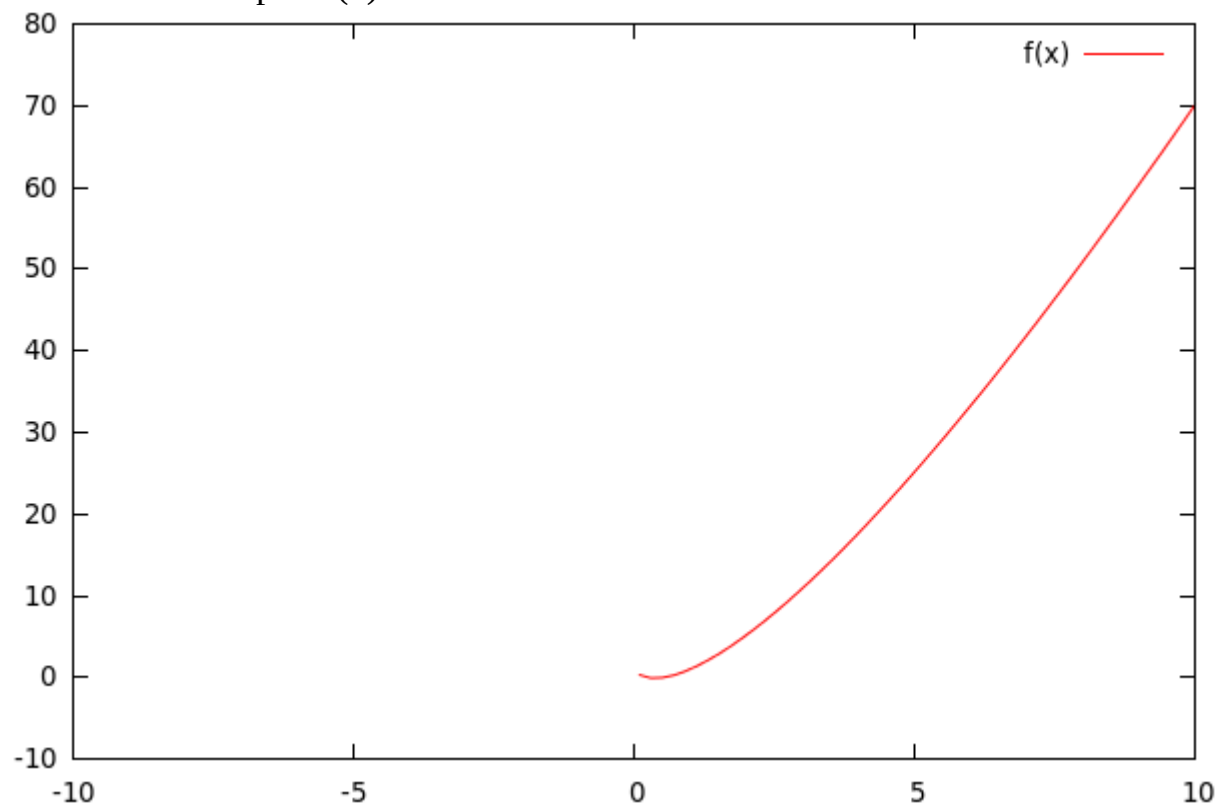
c) Caso Promedio:

- Eficiencia Teórica:

$$T_{1/2}(n) = (7/2)n\log(n) + 1$$

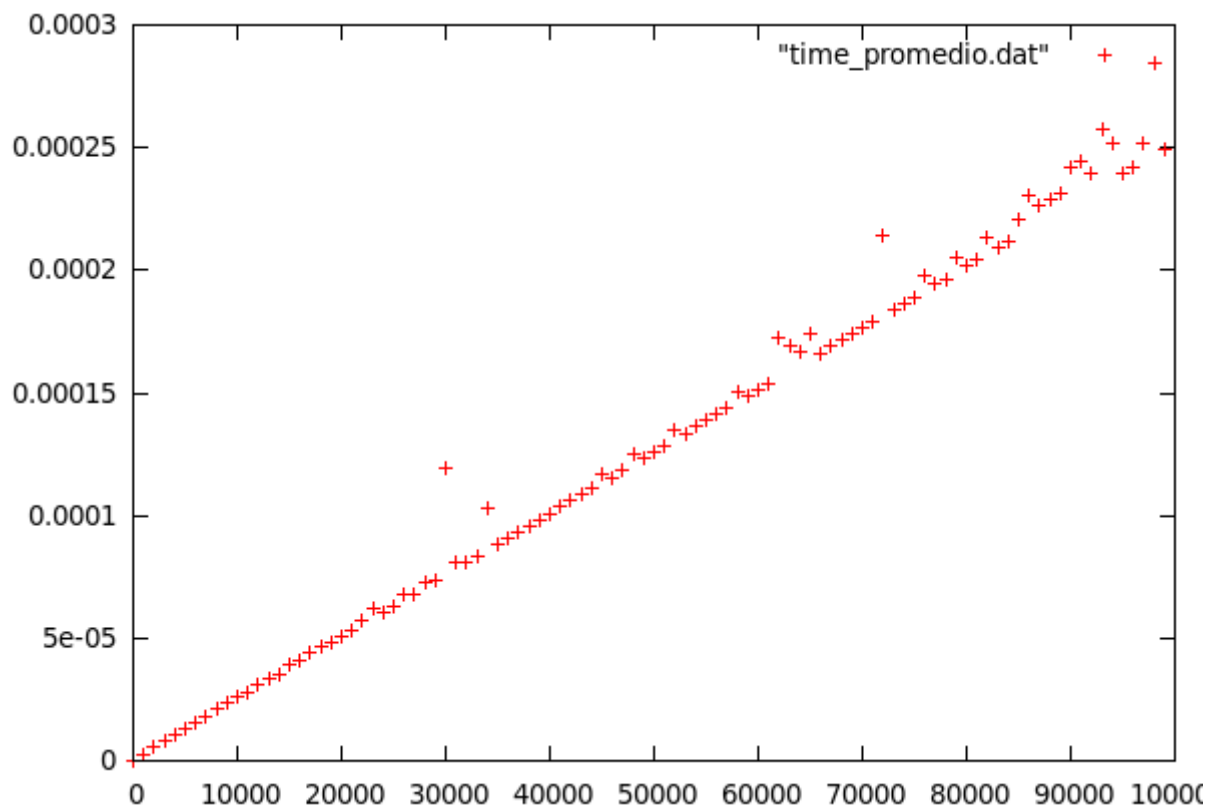
$$f(x) = (7/2)*x*\log(x) + 1$$

plot f(x)



- Eficiencia empírica:

plot "time_promedio.dat"



- Ajuste:

$$f(x) = a \cdot x \cdot \log(x) + b$$

fit $f(x)$ "time_promedio.dat" via a, b

plot "time_promedio.dat", $f(x)$

