ANALISIS DE ALGORITMOS

Equipo:

Jaime Alejandro Salinas Núñez

Martín Eduardo Barriga Vargas

José Manuel Ramírez Vives

3CM3

Reporte practica 1

“Pruebas a posteriori”

4 de marzo de 2019

**PLANTAMIENTO DEL PROBLEMA**

La practica consiste en llevar a cabo pruebas de ordenamiento de diferentes cantidades de números y medir el tiempo que tarda un determinado algoritmo en ordenar dicha cantidad de números.

Se tienen 7 algoritmos de ordenamiento de números los cuales se tuvieron que implementar para poder llevar a cabo las pruebas, siendo los algoritmos los siguientes:

* Burbuja Simple
* Burbuja Optimizada
* Inserción
* Selección
* Shell
* Ordenamiento con árbol binario de búsqueda
* **Heap**

Se cuenta con un archivo el cual contiene 10 millones de numero diferentes, este archivo funciona como la entrada para cada algoritmo.

Para cada algoritmo se usarán diferentes cantidades de números (parámetro n) y el algoritmo deberá de ordenar esa cantidad de números.

El tiempo que cada algoritmo tarde en ordenar el numero dado de números será medido y reflejado en este reporte.

Al igual basándonos en la medición de los tiempos se realizarán aproximaciones a las formas que dan los algoritmos en tiempo mediante polinomios algebraicos para entender mejor el comportamiento que los algoritmos arrojan.

**ACTIVIDADES Y PRUEBAS**

1. **Medir tiempo que cada algoritmo tarda en ordenar 1,000,000 de números**

Para cada algoritmo, se dió una entrada de Un millón de números en desorden y el algoritmo debía de ordenarlos en forma correcta.

El objetivo es medir el tiempo que tarda cada algoritmo en llevar a cabo la ordenación de dicha cantidad de números.

Cabe destacar que se tomó como limite 1,000,000 de números, debido a que ciertos algoritmos no son capaces de ordenar el archivo completo de números ( archivo con 10 millones de números) en un tiempo que no sea excesivo.

Se tiene una tabla (***Tabla 1***) la cual contiene la comparación de los datos medidos de tiempo para cada algoritmo para una entrada de un millón de números.

Siendo los datos que se tienen:

**Tiempo real:** Tiempo total que tardó el algoritmo en ordenar la cantidad de números.

**Tiempo CPU:** Tiempo total que se mantuvo el algoritmo corriendo en procesador.

**Tiempo E/S:** Tiempo total de tiempo con entradas y salidas de datos.

**% CPU/Wall:** Porcentaje que refleja el rendimiento del CPU durante el proceso de ordenación.

| Tabla 1: Comparación de mediciones | | | | |
| --- | --- | --- | --- | --- |
| Algoritmo (S) | Tiempo real  (S x 1000) | Tiempo CPU  (S x 1000) | Tiempo E/S  (S x 1000) | % CPU/Wall (%) |
| Burbuja | 3602163 | 3359839 | 48707.87 | 94.62 |
| Burbuja Optimizada | 3804053 | 3411025 | 28713.61 | 90.42 |
| Inserción | 1054184 | 939689.5 | 8939.68 | 89.98 |
| Selección | 1293015 | 1254371 | 6276.74 | 97.49 |
| Shell | 36066.91 | 33463.52 | 354.67 | 93.76 |
| Tree Sort | 1216.54 | 1163.25 | 0.29 | 99.03 |
| **Heap** | 723.83 | 713.03 | 6.28 | 99.37 |

De igual forma se obtienen 2 gráficas en las cuales podemos ver cómo fue el comportamiento de cada algoritmo en tiempo para ordenar Un millón de números, siendo la ***Gráfica 1*** la gráfica que refleja el tiempo Real del algoritmo y la ***Gráfica 2*** la que refleja el tiempo en CPU del algoritmo.

Cabe destacar que tanto en la ***Tabla 1*** como en la ***Gráfica 1*** y la ***Gráfica 2*** los tiempos que tarda el algoritmo en ordenar los números (Tiempo real y de CPU) están dados en segundos multiplicados por 1,000.

De la ***Tabla 1*** podemos observar cómo el método de ordenamiento **Heap** gana en el menor tiempo (Tiempo CPU) que consume al ordenar el millón de números, seguido del método de ordenamiento por árbol binario seguido del método Shell. De igual forma podemos observar cómo el método de ordenamiento burbuja y burbuja optimizado, tienen tiempos muy muy grandes, excesivos, comparados con **Heap**, Shell, y Tree sort. Seguido de burbuja sigue inserción y selección que podemos notar que tienen tiempos similares, pero también altos.

De la ***Gráfica 1*** y de la ***Gráfica 2*** se ve mejor reflejado lo que se comenta en el párrafo anterior, teniendo mucha más presencia las barras de los métodos de ordenación burbuja. En estas gráficas se puede notar como el método de ordenamiento Tree sort y **Heap** no se alcanzan a distinguir, esto debido a que la escala se vuelve muy grande gracias a los ordenamientos que son muy altos en tiempo de operación.

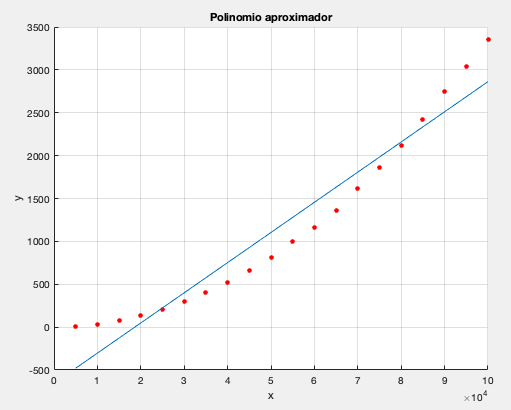
Si redujéramos la escala de la ***Gráfica 1*** y la ***Gráfica 2*** obtendríamos algo parecido a la ***Gráfica 3*** y ***Gráfica 4*** mostradas a continuación.

**6. Realizar una aproximación polinomial del comportamiento temporal (tiempo real), de cada uno de los algoritmos probados según el punto 4.**

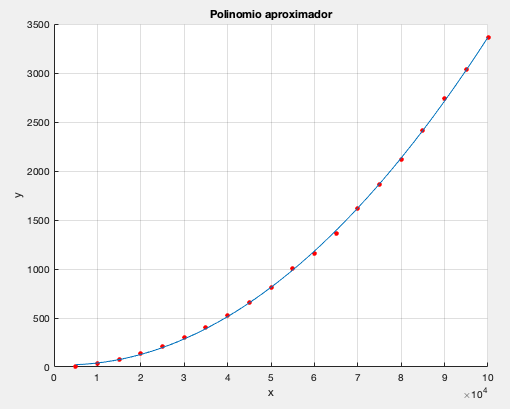
Aproximar cada algoritmo con un polinomio de grado 1, 2, 3, 4 y 8.

Ordenamiento Burbuja

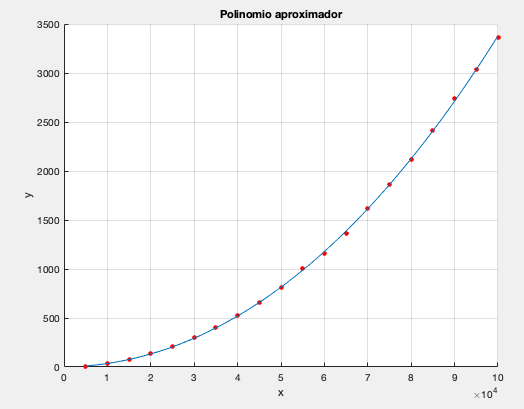
Polinomio grado 1

0.0352 x - 654.8569

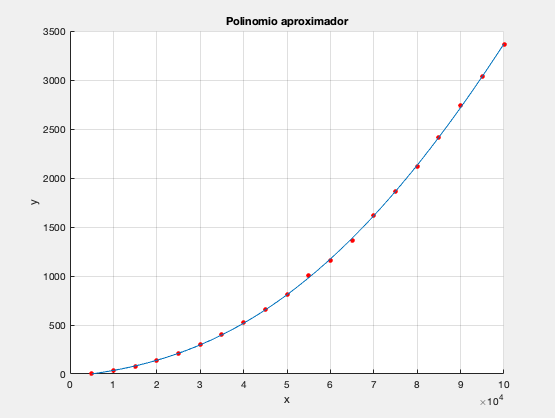
Polinomio grado 2

3.5267e-07 x^2 - 0.0019 x + 24. 0438

Polinomio grado 3

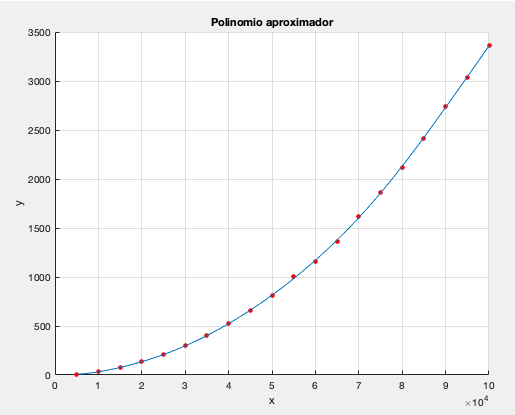
3.7244e-13 x^3 + 2.9401e-07 x^2 + 6.7332-04 x - 0.6915

Polinomio grado 4

-1.0997e-17 x^4 + 2.6820e-12 x^3 + 1.3544e-07 x^2 + 0.0045 x - 25.7337

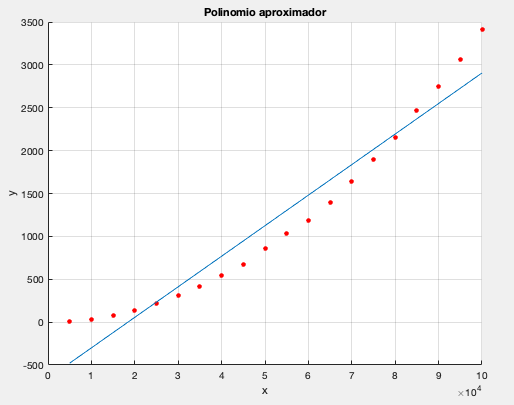
Polinomio grado 8

8.7426e-36 x^8 - 3.6800e-30 x^7 + 6.3027e-25 x^6 - 5.6883e-20 x^5 + 2.9335e-15 x^4 - 8.719906774793458e-11 x^3 + 1.7376 e-06 x^2 - 0.0105 x + 27.0789



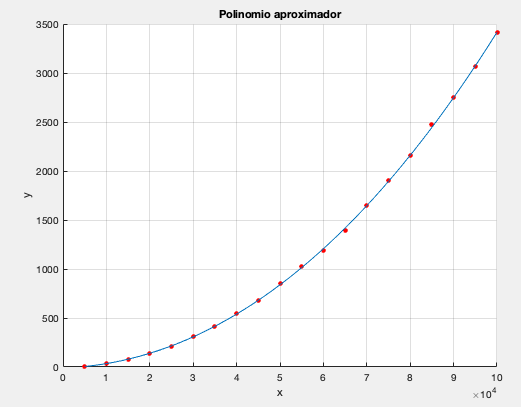
Ordenamiento Burbuja Optimizada

Polinomio grado 1

0.0356 x - 6.5625e+02 

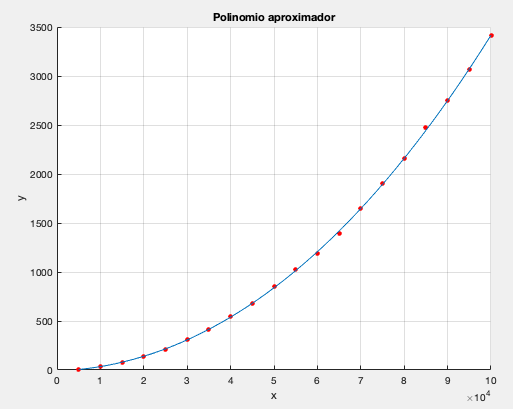
Polinomio grado 2

3.4848e-07 x^2 - 9.6878e-04 x + 14.5835



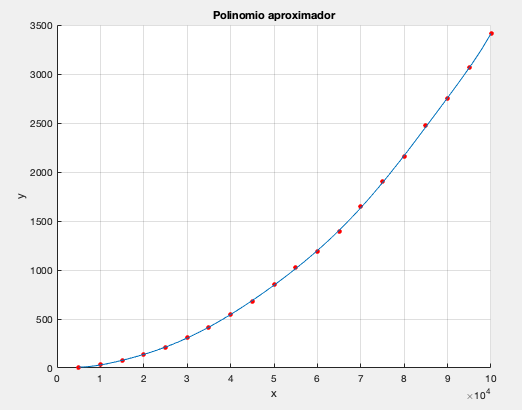
Polinomio grado 3

3.7244e-13 x^3 + 2.8982e-07 x^2 + 0.0015 x - 10.151256996697589



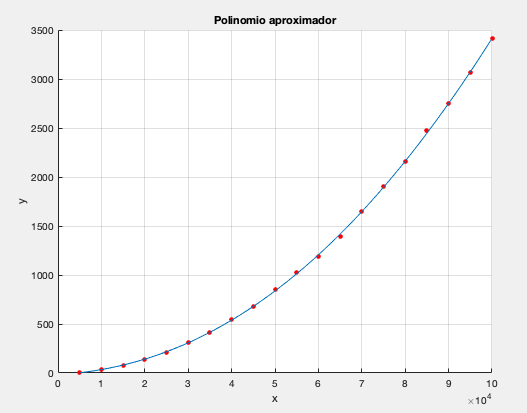
Polinomio grado 4

-2.9631e-18 x^4 + 9.9469e-13 x^3 + 2.4710e-07 x^2 0.0026 x -16.8982



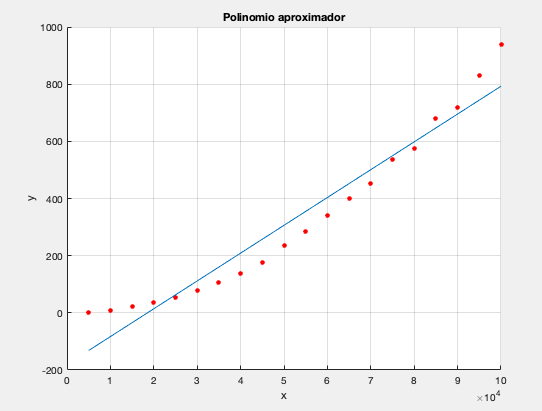
Polinomio grado 8

2.7214e-35 x^8 - 1.083e-29 x^7 + 1.7610e-24 x^6 - 1.5058e-19 x^5 + 7.3132e-15 x^4 - 2.0340e-10 x^3 + 3.4342e-06 x^2 - 0.0226 x + 57.2786

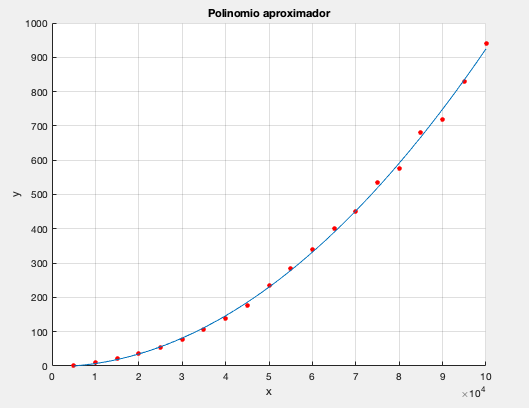


Ordenamiento Inserción

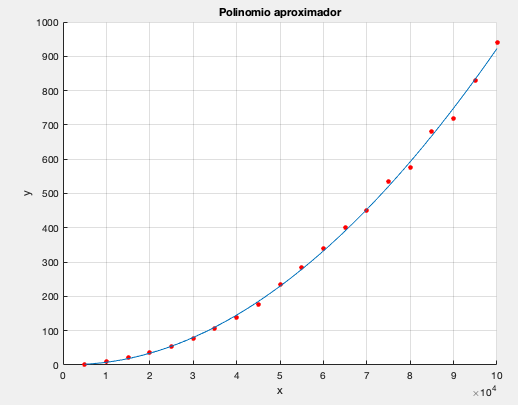
Polinomio grado 1

0.0097 x - 1.8040e+02

Polinomio grado 2

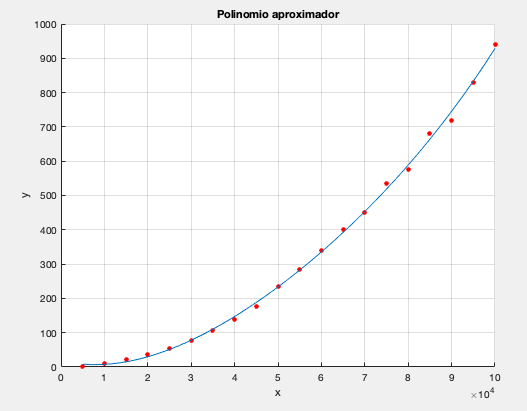
9.2522e-08 x^2 + 1.8057e-05 x - 2.2954

Polinomio grado 3

-5.7265e-14 x^3 + 1.0154e07 x^2 + 3.7005e04 x + 1.5077

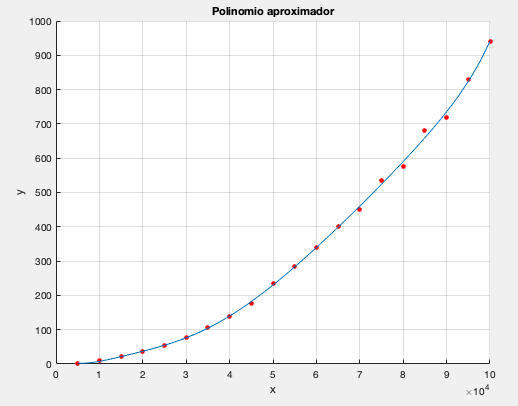
Polinomio grado 4

7.2885e-18 x^4 - 1.5878e-12 x^3 + 2.0662e-07 x^2 + 0.0029x + 18.1037



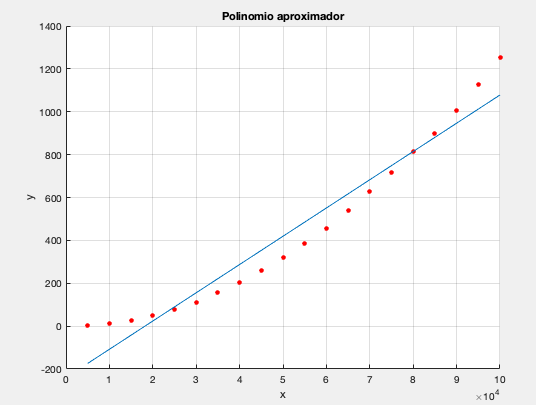
Polinomio grado 8

6.75048e-36 x^8 - 2.8591e-30 x^7 + 5.0748e-25 x^6 - 4.8450e-20 x^5 + 2.65046e-15 x^4 - 8.1996e-11 x^3 + 1.4319e-06 x^2 - 0.0102 x + 26.4869



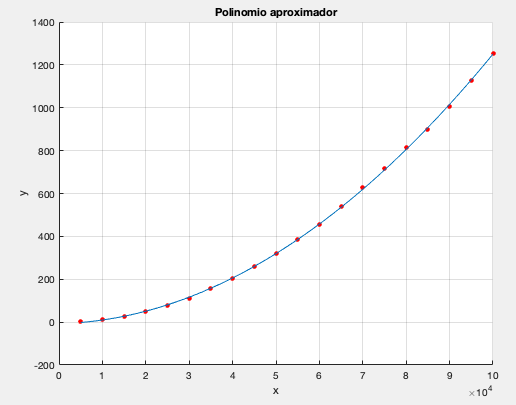
Ordenamiento Selección

Polinomio grado 1

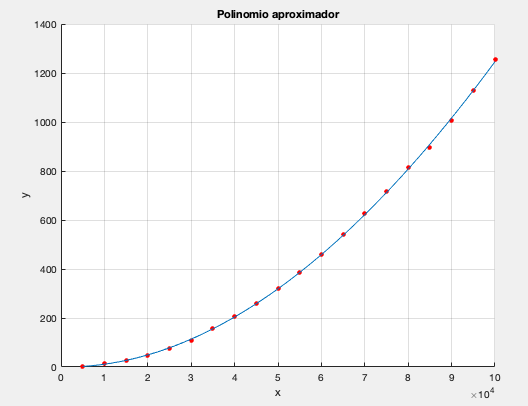
0.0132 x - 239.7954

Polinomio grado 2

1.2074e-07 x^2 + 5.0340-04 x - 7.3598

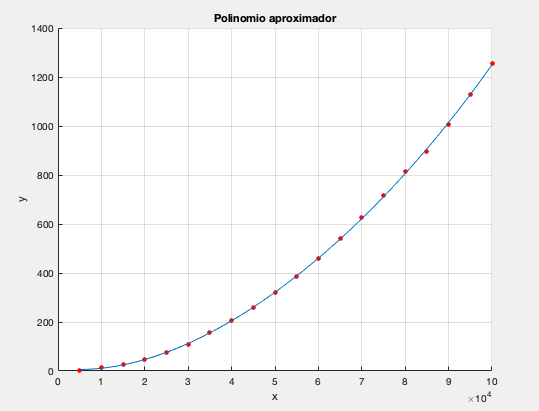


Polinomio grado 3

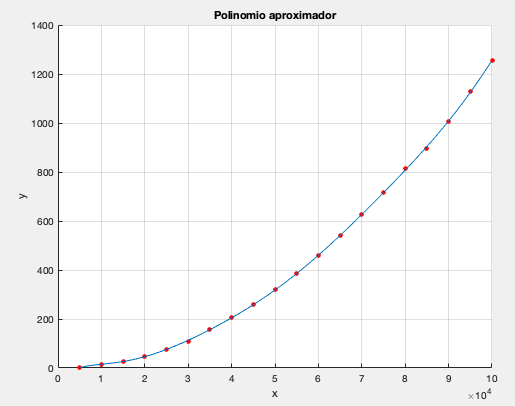
-1.1372e-13 x^3 + 1.3865e-07 x^2 - 2.6737e-04 x + 0.1930

Polinomio grado 4

4.3716e-18 x^4 - 1.0317e-12 x^3 + 2.0168-07 x^2 - 0.0018 x + 10.1471



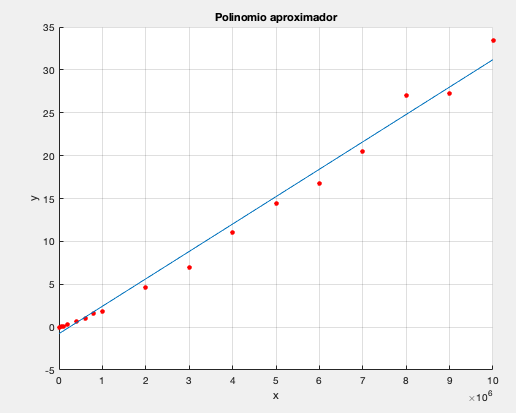
Polinomio grado 8

-9.0647e-36 x^8 + 3.9709e-30 x^7 -7.1416e-25 x^6 + 6.8143e-20 x^5 - 3.72049e-15 x^4 + 1.1657e-10 x^3 -1.8519e-06 x^2 + 0.0157 x- 42.5920

Ordenamiento Shell

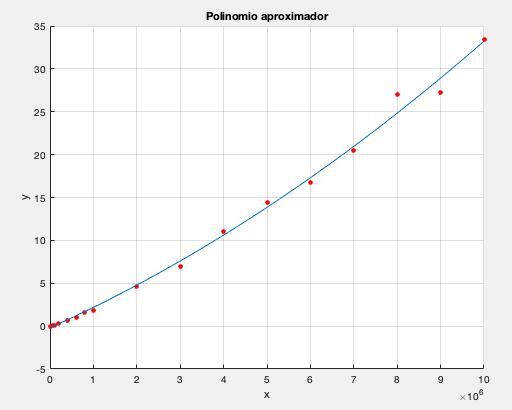
Polinomio grado 1

3.1954e-06 x - 0.7592



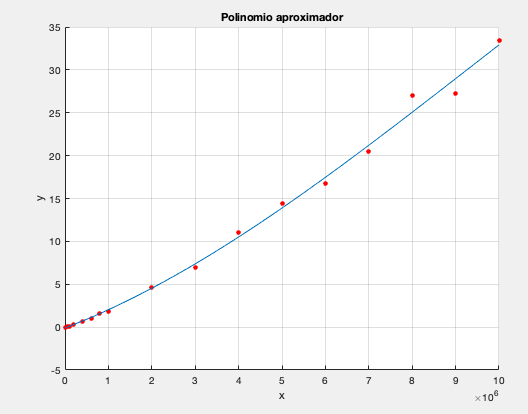
Polinomio grado 2

1.0664e-13 x^2 2.2755e-06 x - 0.1893

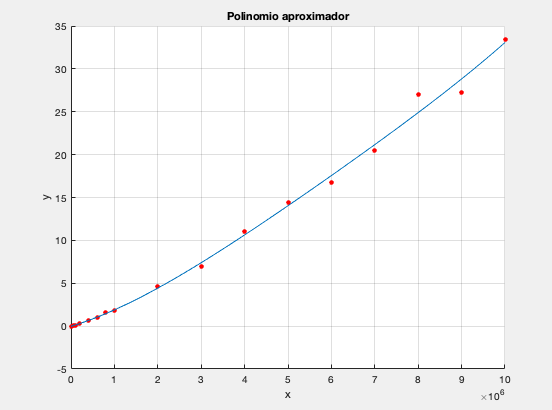


Polinomio grado 3

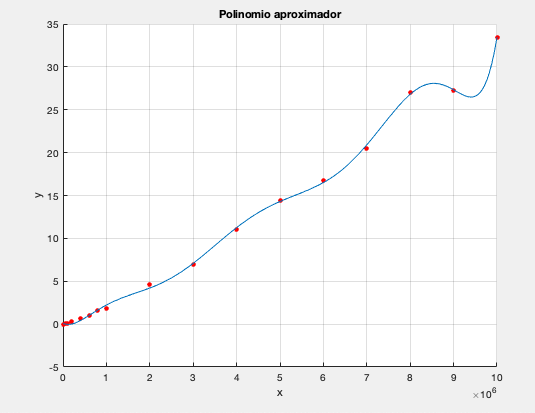
-7.5449e-21 x^3 + 2.1403e-13 x^2 + 1.9142e-06 x - 0.0888



Polinomio grado 4

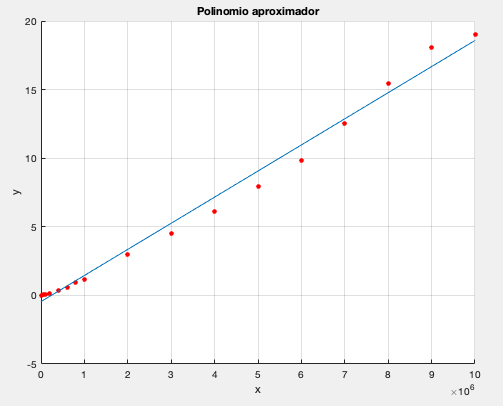
1.8745e-27 x^4 - 4.3816-20 x^3 + 4.2848e-13 x^2 + 1.5320e-06 x - 0.0244

Polinomio grado 8

2.9310e-52 x^8 - 1.0785e-44 x^7 + 1.5907e-37 x^6 - 1.2014e-30 x^5 + 4.9225e-24 x^4 - 1.0675e-17 x^3 + 1.1222e-11 x^2 - 2.2845e-06 x + 0.0975

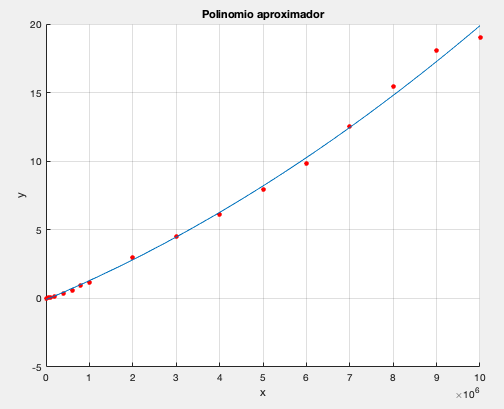
Ordenamiento Árbol Binario

Polinomio grado 1

1.9032e-06 x - 0.4503

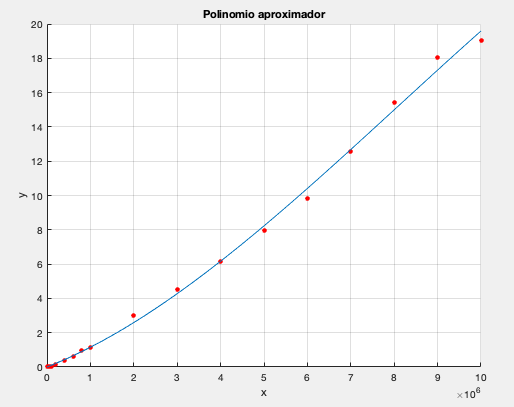
Polinomio grado 2

6.7982e-14 x^2 + 1.3168e-06 x - 0.0871



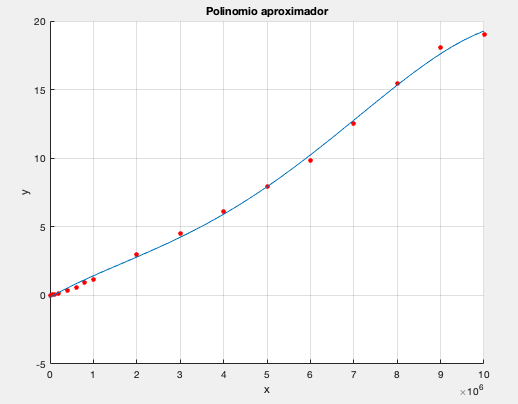
Polinomio grado 3

-6.7736e-21 x^3 + 1.6439e-13 x^2 + 9.9249e-07 x + 0.0030



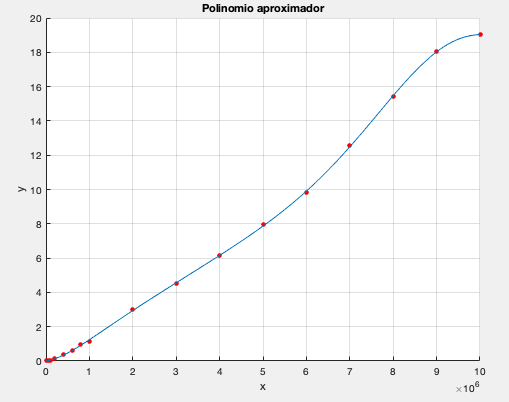
Polinomio grado 4

-3.7967e-27 x^4 + 6.6691e-20 x^3 - 2.6993e-13 x^2 + 1.7665e-06 x - 0.1274



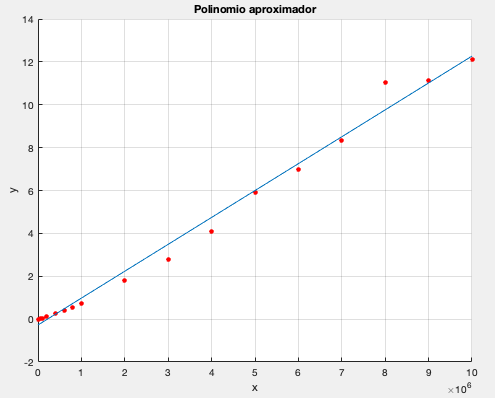
Polinomio grado 8

1.1161e-53 x^8 - 4.3608e-46 x^7 + 6.9273e-39 x^6 - 5.8968e-32 x^5 + 2.9823e-25 x^4 - 9.0923e-19 x^3 + 1.56826e-12 x^2 + 3.3094e-07 x + 0.0056



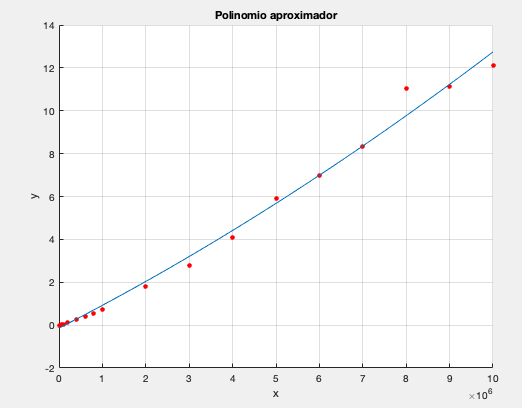
**Ordenamiento Heap**

Polinomio grado 1

1.2543e-06 x - 0.2729

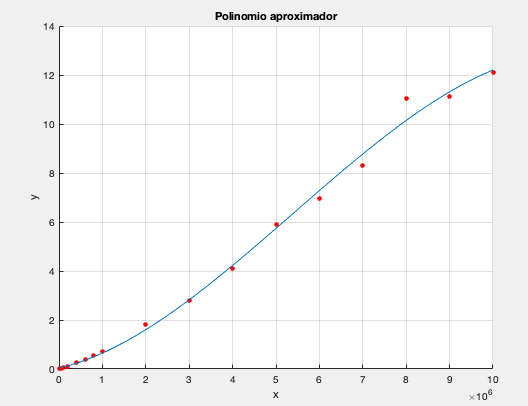
Polinomio grado 2

2.4989e-14 x^2 + 1.0388e-06 x - 0.1393



Polinomio grado 3

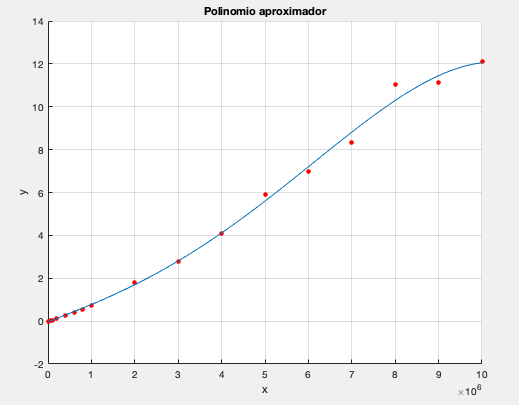
-1.2886e-20 x^3 + 2.0841e-13 x^2 + 4.2166e-07 x + 0.0321



Polinomio grado 4

-1.7298e-27 x^4 + 2.0584e-20 x^3 + 1.0531e-14 x^2 + 14 7.7432e-07 x

- 0.0272



Polinomio grado 8

1.067e-52 x^8 - 3.9977e-45 x^7 + 6.0218e-38 x^6 - 4.6655e-31 x^5 + 1.9717e-24 x^4 - 4.4346e-18 x^3 + 4.8438e-12 x^2 - 1.13455e-06 x + 0.0522

