

Los tiempos están en segundos (s)

N	Bubble sort		Cocktail sort		Insert sort		Bucket sort		Counting sort
	PC	BBB	PC	BBB	PC	BBB	PC	BBB	PC
10	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0
1 000	0.0089	0.009	0.000006	.000007	0.00043	0.0006	0	0	0
10 000	0.212542	0.3536	0.000036	.000215	0.0034	0.0189	0	0	0
100 000	16.359367	25.72	0.000342	0.00345	0.04328	0.6125	0.001	0.05	0.002
1 000 000	1632.003	Inf	5.169082	inf	0.60989	inf	0.0256	0.17	0.0675

N	Merge sort		Tree sort		Radix		Shell		Select sort	
	PC	BBB	PC	BBB	PC	BBB	PC	BBB	PC	
10	0	0	0	No corrio	0	0	0	0	0	
100	0	0	0		0	0	0	0	0.001	
1 000	0	0	0.002		0.005	0	0.00156	0.00001	0.074	
10 000	0.003384	0.05356	0.238561		0.00321	0.06	0.02654	0.0028	0.2318	
100 000	1.526	0.32	0.837		0.343	0.39	137.951	638.43	570.812	
1 000 000	12.3656	2.29	4.15613		2.1563	2.27	1024.15	inf	inf	

Juan Carlos León Jarquín  
A01020200

Nos damos cuenta en una máquina, en este caso un BeagleBone Black, que tiene los recursos limitados, se tarda más en correr, ya que a pesar de usar libremente los recursos, son tan limitados que corren lentamente.

En cuanto a correr los algoritmos con cantidades de millones produce en algunos casos un StackOverflow dentro de la laptop, mientras que en la BeagleBone, simplemente ninguno pudo completar la acción.

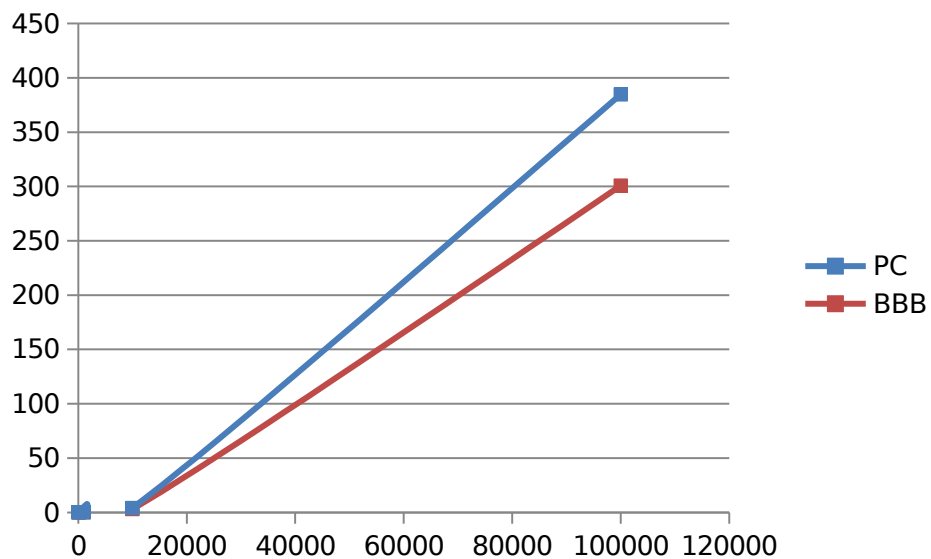
También con los datos nos damos cuenta que BucketSort fue el algoritmo que mejor rendimiento tuvo dentro de la laptop, en cuanto a la BeagleBone nos damos cuenta que es Counting Sort.

Como en clases pasadas vimos, el peor rendimiento lo tiene el bubble sort, ya que hace demasiadas comparaciones, lo cual implica más tiempo de ejecución

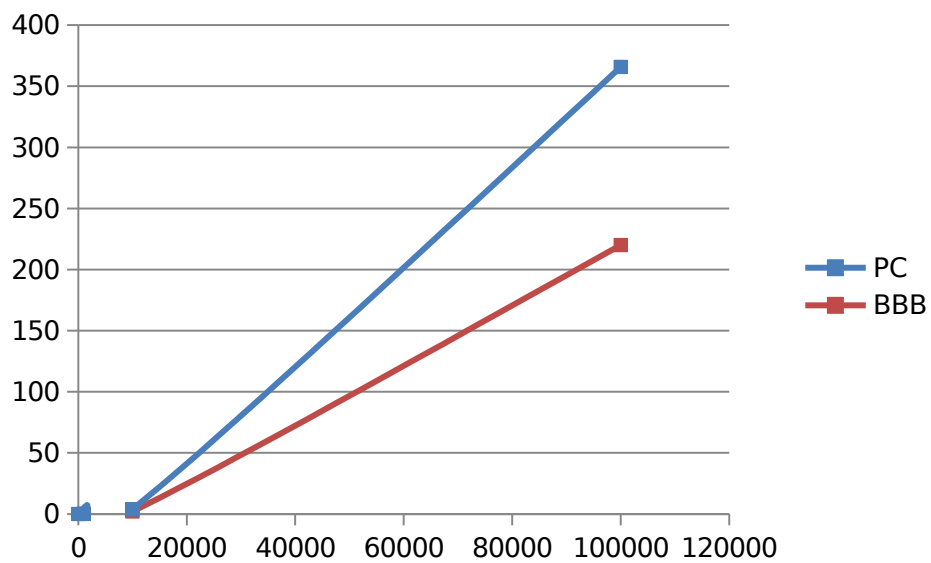
Juan Carlos León Jarquín  
A01020200

## Tablas comparativas

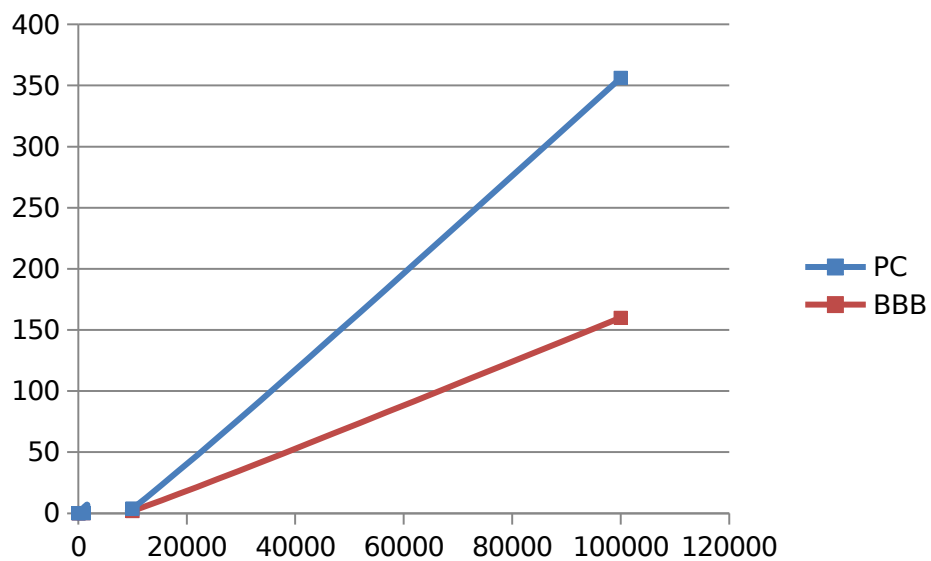
Bubble sort:



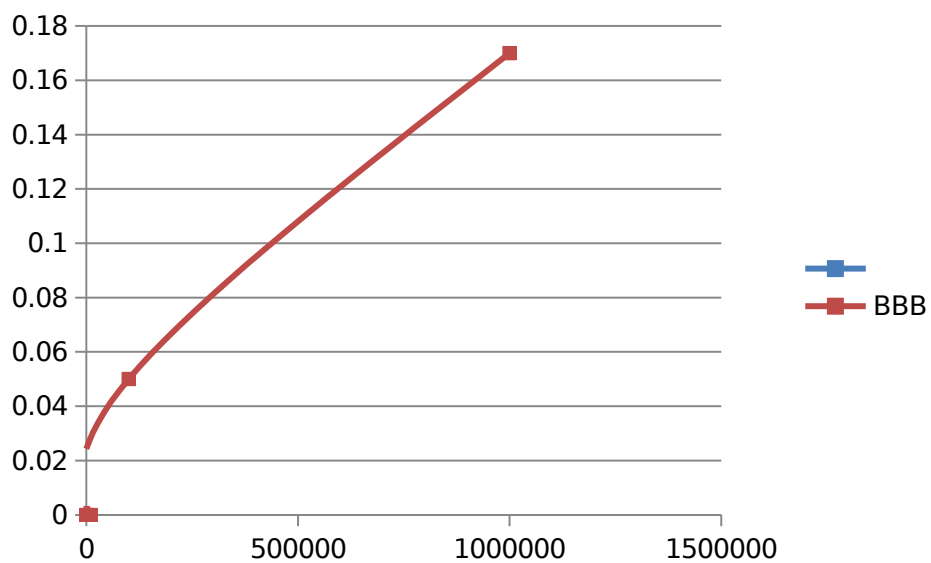
Quick Sort:



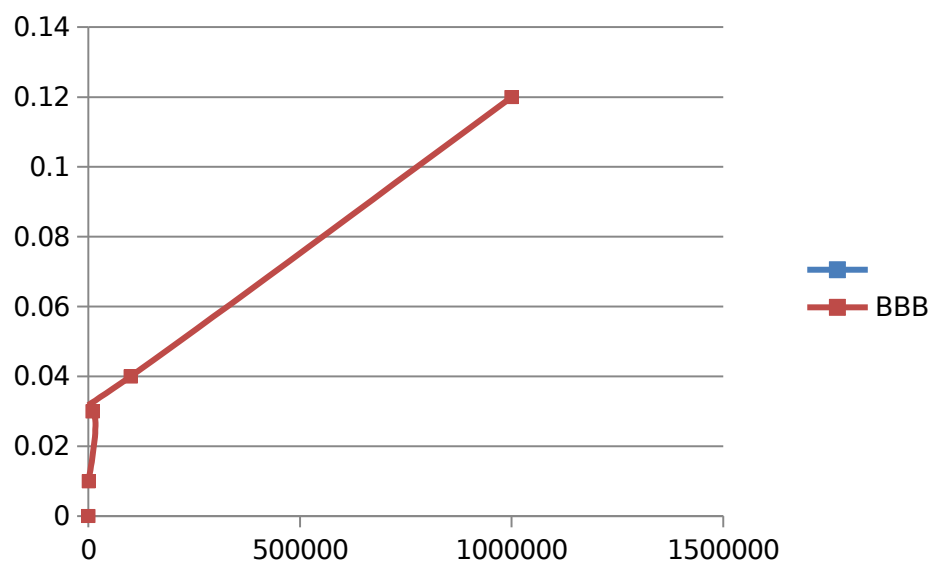
### Insert Sort:



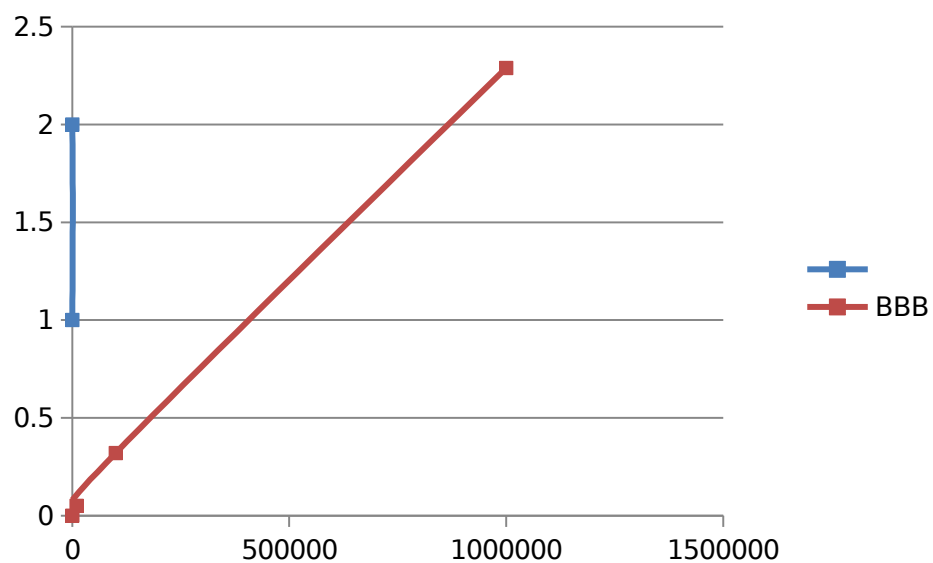
### Bucket sort:



Counting sort:

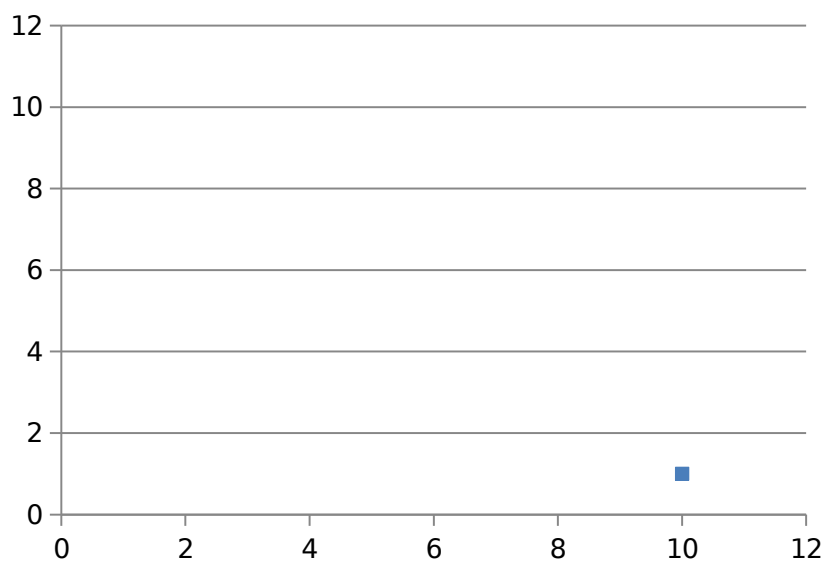


Merge sort:

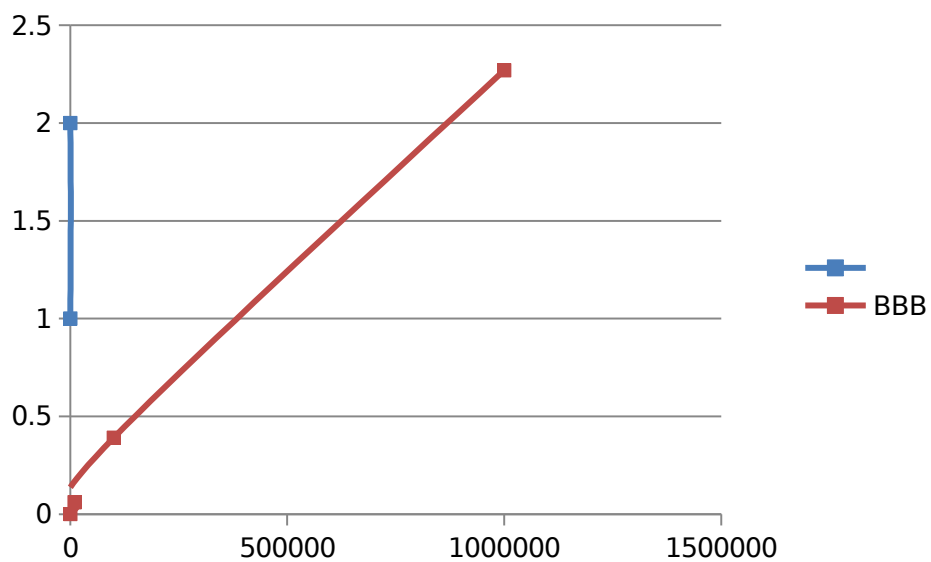


Juan Carlos León Jarquín  
A01020200

Binary Tree sort:

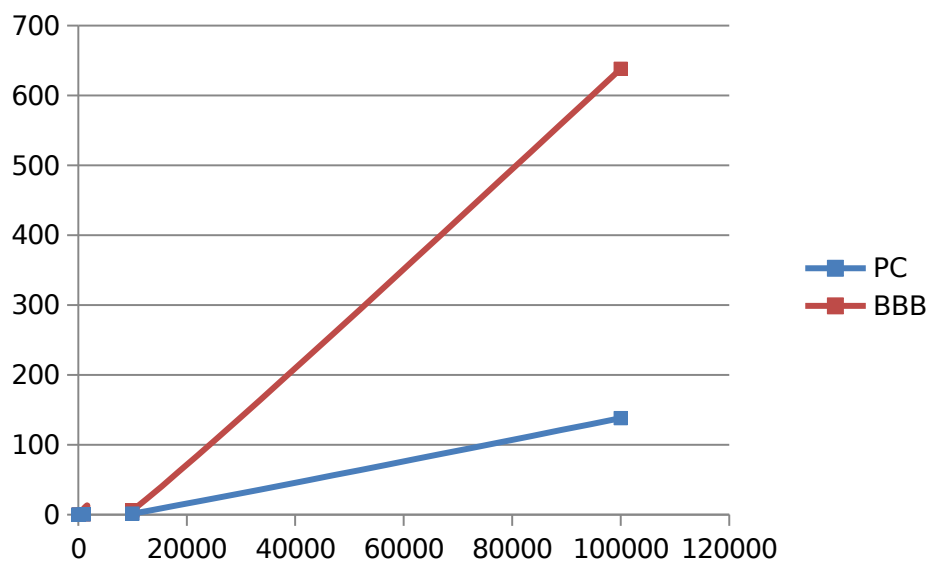


Radix sort:

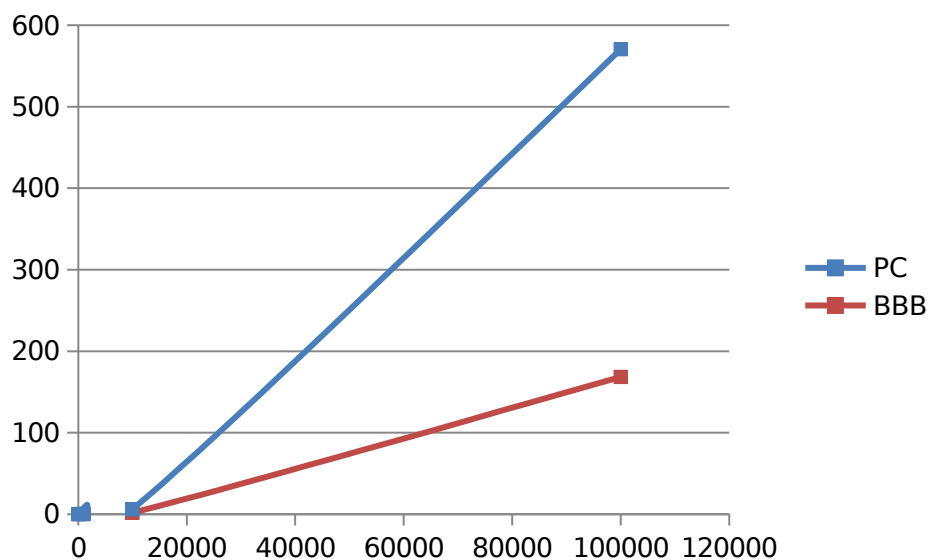


Juan Carlos León Jarquín  
A01020200

Shell sort:

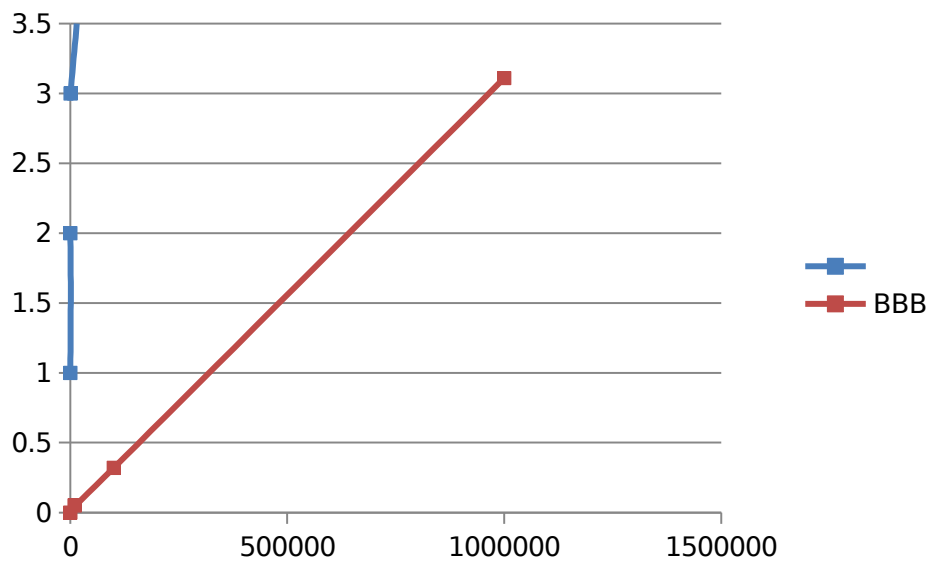


Select sort:



Juan Carlos León Jarquín  
A01020200

Heap sort:



Quick sort:

