

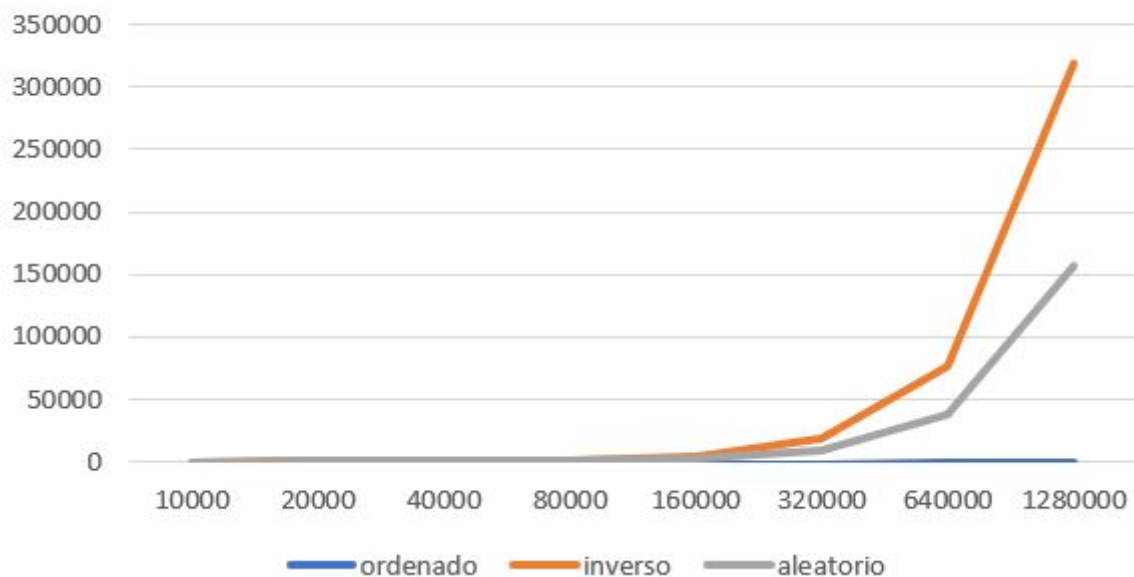
# PRÁCTICA 2

Ángela López López UO270318

## INSERCIÓN

N	t ordenado (ms)	t inverso (ms)	t aleatorio (ms)
10000	0,016	62	47
20000	0,016	251	125
40000	0,016	284	126
80000	0,031	1135	531
160000	0,062	4649	2249
320000	0,078	18844	9380
640000	0,188	76020	37674
1280000	0,422	319523	156680

### Inserción

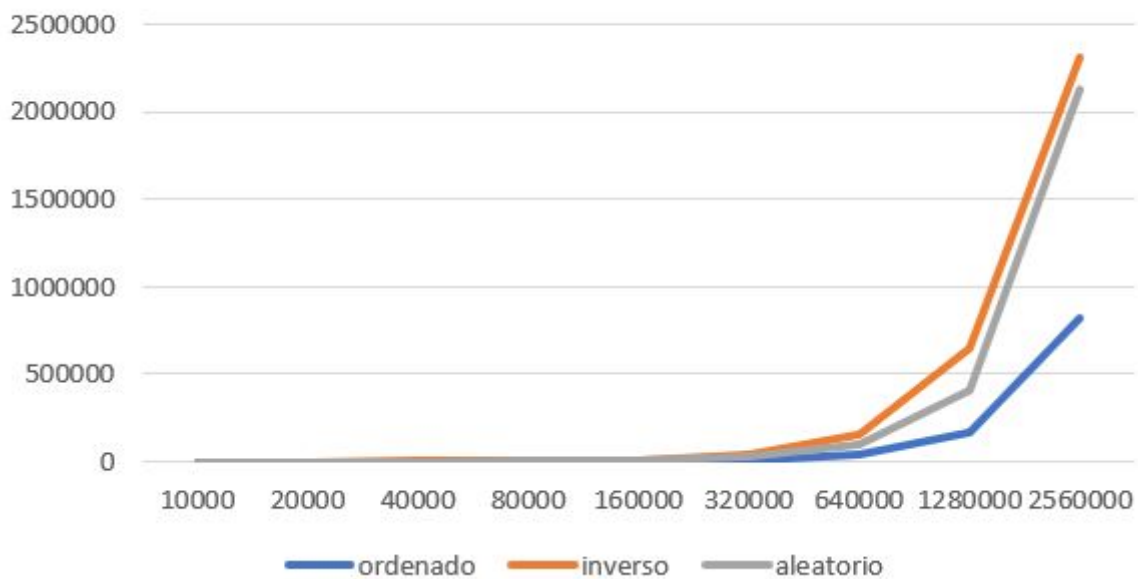


- MEJOR: ordenado  $\rightarrow O(n)$ , los elementos del vector ya están ordenados
- PEOR: inverso  $\rightarrow O(n^2)$ , todos los elementos del vector necesitan ser reordenados.
- MEDIO: aleatorio  $\rightarrow O(n^2)$ , algunos elementos del vector necesitan ser reordenados.

## SELECCIÓN

N	t ordenado (ms)	t inverso (ms)	t aleatorio (ms)
10000	31	56	32
20000	32	171	94
40000	172	694	399
80000	593	2545	1583
160000	2446	10423	6357
320000	9908	38638	24998
640000	40202	155610	99846
1280000	170490	646091	406564
2560000	820344	2316930	2126636

### Selección

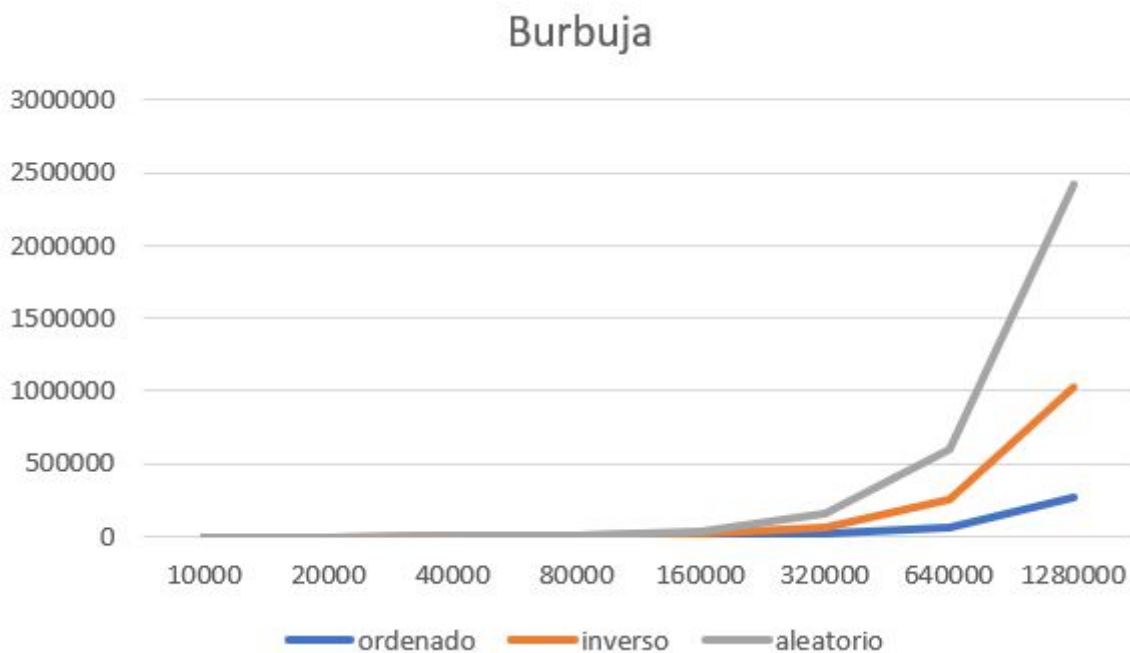


- MEJOR: ordenado  $\rightarrow O(n^2)$
- PEOR: inverso  $\rightarrow O(n^2)$
- MEDIO: aleatorio  $\rightarrow O(n^2)$

Todos los casos tienen una complejidad cuadrática, ya que el algoritmo necesita encontrar la posición del más pequeño  $O(n)$  e intercambiar:  $O(n) * O(n) == O(n^2)$ .

## BURBUJA

N	t ordenado (ms)	t inverso (ms)	t aleatorio (ms)
10000	15	78	141
20000	64	250	547
40000	219	969	2171
80000	962	3999	9357
160000	3987	15918	37913
320000	16754	63313	152121
640000	64739	253628	600125
1280000	267822	1029566	2413264



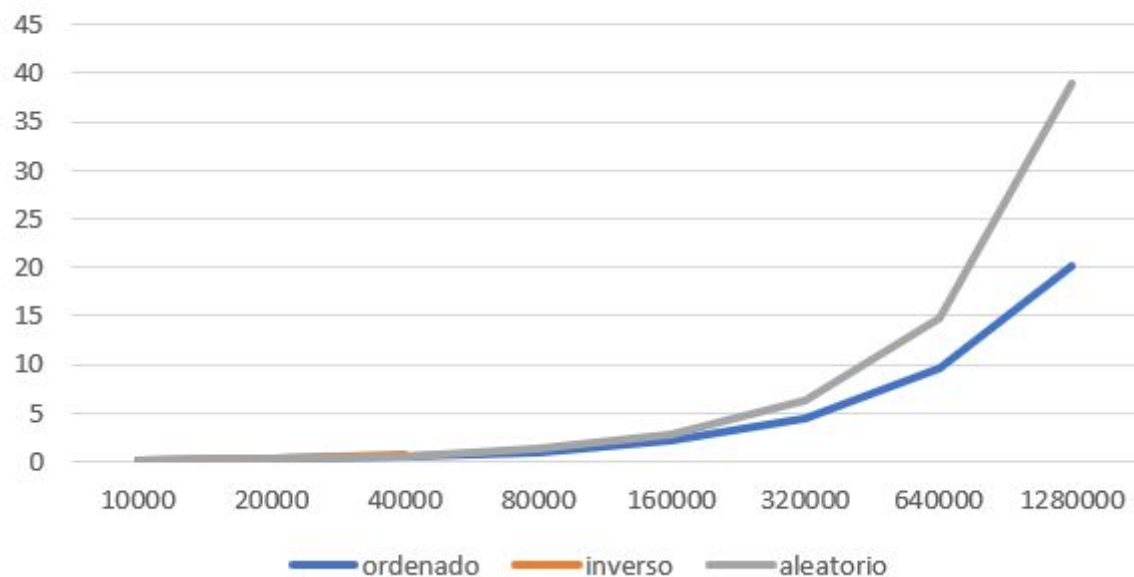
- MEJOR: ordenado  $\rightarrow O(n^2)$
- PEOR: aleatorio  $\rightarrow O(n^2)$
- MEDIO: inverso  $\rightarrow O(n^2)$

Todos los casos tienen una complejidad cuadrática, ya que tiene que iterar por todo el array.

## QUICKSORT (mediana a 3)

N	t ordenado (ms)	t inverso (ms)	t aleatorio (ms)
10000	0,109	0,172	0,14
20000	0,234	0,313	0,281
40000	0,485	0,734	0,594
80000	1,031		1,266
160000	2,141		2,75
320000	4,552		6,209
640000	9,56		14,856
1280000	20,05		38,865

Quicksort



INVERSO: Stackoverflow a partir de  $n=40000$

- MEJOR: ordenado  $\rightarrow O(n \log n)$
- PEOR: aleatorio  $\rightarrow O(n \log n)$
- MEDIO: inverso  $\rightarrow O(n \log n)$

Es el mejor algoritmo de los 4 vistos. Se usa como pivote la mediana de una muestra de 3 elementos, por lo que al no usar el elemento central como pivote el caso de ordenado no es el de peor complejidad.

## CONCLUSIONES

Viendo la gráfica de tiempos se puede observar como quicksort es mucho más rápido que el resto de algoritmos que hemos utilizado, ya que quitando el método de inserción con un vector ordenado, todos tienen complejidad cuadrática.

## OPTATIVO

Para  $n=80 \cdot 10^6$  calcular aleatorio de inserción, selección, burbuja, rápido central y rápido mediana 3.

$k=1000 \rightarrow O(n^2)$

- BURBUJA:  $(10^3)^2 \cdot 9 \text{ sec} = 9 \text{ millones sec} = 100 \text{ días}$
- SELECCION:  $1.500.000 \text{ sec} = (10^3)^2 \cdot 1,5 \text{ sec} = 1,5 \text{ millones sec} = 17 \text{ días}$
- INSERCIÓN:  $(10^3)^2 \cdot 0,5 \text{ s} = 500.000 \text{ sec} = 5 \text{ días}$
- RÁPIDO CENTRAL:  $(10^3) \log(10^3) \cdot 0,01 \text{ sec} = 99 \text{ sec} = 0.0011 \text{ días}$
- RÁPIDO MEDIANA 3:  $(10^3) \log(10^3) \cdot 0,01 \text{ s} = 99 \text{ sec} = 0.0011 \text{ días}$