Algorithmics	Student information	Date	Number of session
	UO: UO294515	24/02/2025	6
	Surname: Lopez Garcia		
	Name: Oscar		

Escuela de Ingeniería Informática
Universidad de Oviedo

Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Activity 1. [Direct exchange or Bubble algorithm]

n	T ordered	T reverse	T random
10000	333	1524	1079
2*10000	1254	5976	4265
2^2*10000	4908	23844	16829
2^3*10000	19757	OoT	67368
2^4*10000	OoT	OoT	OoT

All three, ordered, reverse and random have a quadratic complexity O(n^2), the times make sense as reverse has to make more changes in the vector than random, and ordered has no changes to do.

Activity 2. [Selection algorithm]

n	T ordered	T reverse	T random
10000	313	296	320
2*10000	1260	1176	1266
2^2*10000	5045	4507	5022
2^3*10000	19450	18041	20069
2^4*10000	OoT	OoT	OoT

All three, ordered, reverse and random have a quadratic complexity O(n^2). It makes sense as for this method it is not better if the vector is ordered or if it is not.

Algorithmics	Student information	Date	Number of session
	UO: UO294515	24/02/2025	6
	Surname: Lopez Garcia		
	Name: Oscar		

Activity 3. [Insertion algorithm]

n	T ordered	T reverse	T random
10000	LoR	299	154
2*10000	LoR	1158	584
2^2*10000	LoR	4683	2344
2^3*10000	LoR	18816	9277
2^4*10000	LoR	74765	37072
2^5*10000	LoR	OoT	OoT
2^6*10000	LoR	OoT	OoT
2^7*10000	LoR	OoT	OoT
2^8*10000	LoR	OoT	OoT
2^9*10000	91	OoT	OoT
2^10*10000	179	OoT	OoT
2^11*10000	359	OoT	OoT
2^12*10000	728	OoT	OoT
2^13*10000	1452	OoT	OoT

Yes, as the complexity is linear O(n) for the ordered vectors as it does not have to enter the while loop, and it is quadratic O(n^2) for the reverse and random vectors, as it enters the while loop that iterates n times.

Algorithmics	Student information	Date	Number of session
	UO: UO294515	24/02/2025	6
	Surname: Lopez Garcia		
	Name: Oscar		

Activity 4. [Quicksort algorithm]

n	T ordered	T reverse	T random
25000	LoR	LoR	93
2*25000	63	72	192
2^2*25000	123	140	403
2^3*25000	257	289	869
2^4*25000	530	599	1873
2^5*25000	1098	1249	4229
2^6*25000	2263	2523	10312

It matchs the expected as for all cases the complexiity is between nlogn and n^2 which are the best and worse cases of this algorithm.

Buble: O(n^2) for random

n_1 = 10000 -> t_1 = 1079 ms , n_2=16*10^6 -> t_2=?

K=(n_2^2/n_1^2)

t_2 = k*t1 = ((16*10^6)^2/10000^2)*1079 ms = (16*10^{12}/1*10^8)*1079 ms = 16*10^4*1079 ms = 172640000 ms * (1 s/1000ms)*(1h/3600s)*(1 day/24h) = 1,998 days

Selection: O(n^2) for random

n_1 = 10000 -> t_1 = 320 ms , n_2=16*10^6 -> t_2=?

K=(n_2^2/n_1^2)

t_2= k*t1 = ((16*10^6)^2/10000^2)*320 ms = (16*10^{12}/1*10^8)*320 ms = 16*10^4*320 ms = 51200000 ms * (1 s/1000ms)*(1h/3600s)*(1 day/24h) = 0,593 days

Algorithmics	Student information	Date	Number of session
	UO: UO294515	24/02/2025	6
	Surname: Lopez Garcia		
	Name: Oscar		

Selection: O(n^2) for random

n_1 = 10000 -> t_1 = 154 ms , $n_2=16*10^6$ -> t_2=?

$K=(n_2^2/n_1^2)$

t_2 = k*t1 = $((16*10^6)^2/10000^2)*154$ ms = $(16*10^{12}/1*10^8)*154$ ms = $16*10^4*154$ ms = 24640000 ms * (1 s/1000ms)*(1h/3600s)*(1 day/24h) = 0,285 days

Activity 5. [Quicksort+Insertion algorithm]

N = 16*10^6	T random
Quicksort	2766
Quicksort+Insertion (k=5)	2544
Quicksort+Insertion (k=10)	2425
Quicksort+Insertion (k=20)	2377
Quicksort+Insertion (k=30)	2353
Quicksort+Insertion (k=50)	2294
Quicksort+Insertion (k=100)	2133
Quicksort+Insertion (k=200)	1810
Quicksort+Insertion (k=500)	2394
Quicksort+Insertion (k=1000)	4034

We can conclude that this algorithm improves the performance of the quicksort algorithm if we choose a good k , if k is too small, the algorithm does not change too much, and if it is too big, we can get a worse time than if we didn´t implement the insertion, as we will be doing too many sortings with the insertion.

Algorithmics	Student information	Date	Number of session
	UO: UO294515	24/02/2025	6
	Surname: Lopez Garcia		
	Name: Oscar		