| Algorithmics | Student information | Date | Number of session |
|---|---|---|---|
| | UO: UO301879 | 20/02/2025 | 2 |
| | Surname: Sariego Sánchez | | |
| | Name: Martín | | |

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 | 319 | 1506 | 1034 |
| 2*10000 | 1288 | 5903 | 4138 |
| 4*10000 | 5218 | 23666 | 16500 |
| 8*10000 | 20538 | OoT | OoT |
| 16*10000 | OoT | OoT | OoT |

Bubble algorithm always has a complexity of O(n^2), and these times make sense given that. The difference in time between ordered, reverse and random is caused by the vectors used, the ordered vector doesn't need any exchanges, so it takes less.

# Activity 2. Selection algorithm

| n | t ordered | t reverse | t random |
|---|---|---|---|
| 10000 | 310 | 286 | 317 |
| 2*10000 | 1214 | 1116 | 1231 |
| 4*10000 | 4846 | 4508 | 4920 |
| 8*10000 | 19538 | 17942 | 19454 |
| 16*10000 | OoT | OoT | OoT |

The selection algorithm also has a complexity of O(n^2) in all cases, but it is more effective than bubble in the reverse and random vectors due to how it works.

| Algorithmics | Student information | Date | Number of session |
|---|---|---|---|
| | UO: UO301879 | 20/02/2025 | 2 |
| | Surname: Sariego Sánchez | | |
| | Name: Martín | | |

# Activity 3. Insertion algorithm

| n | t ordered | t reverse | t random |
|---|---|---|---|
| 10000 | LoR | 289 | 147 |
| 2*10000 | LoR | 1135 | 570 |
| 4*10000 | LoR | 4548 | 2317 |
| 8*10000 | LoR | 18117 | 9124 |
| 16*10000 | LoR | Oot | 36657 |
| 32*10000 | LoR | Oot | Oot |
| 64*10000 | LoR | Oot | Oot |
| 128*10000 | LoR | Oot | Oot |
| 256*10000 | LoR | Oot | Oot |
| 512*10000 | 96 | Oot | Oot |
| 1024*10000 | 190 | Oot | Oot |
| 2048*10000 | 372 | Oot | Oot |
| 4096*10000 | 739 | Oot | Oot |
| 8192*10000 | 1482 | Oot | Oot |

Unlike he previous two, Insertion algorithm has a best case complexity of O(n), which is why the ordered times are so low. The worst case is still O(n^2), so that's why the other two vectors take longer.

# Activity 4. Quicksort algorithm

| n | t ordered | t reverse | t random |
|---|---|---|---|
| 250000 | LoR | LoR | 98 |
| 2*250000 | 62 | 73 | 209 |
| 4*250000 | 128 | 148 | 408 |
| 8*250000 | 255 | 301 | 910 |
| 16*250000 | 543 | 608 | 1947 |
| 32*250000 | 1121 | 1233 | 4379 |

| | | | |
|---|---|---|---|
| 64*250000 | 2267 | 2525 | 10583 |

Quicksort algorithm has a best case complexity of O(nlogn) and a worst case of O(n^2), but the average case is O(nlogn), making it the least complex until now, as seen in the above times.

To sort 16 million randomly ordered items,

Bubble would take around 30 days and 13 hours.

Selection would take around 9 days and 2 hours.

Insertion would take around 4 days and 6 hours.

## Activity 5. Quicksort + Insertion algorithm

| n | t random |
|---|---|
| Quicksort | 10488 |
| Quicksort+Insertion(k=5) | 10977 |
| Quicksort+Insertion(k=10) | 10825 |
| Quicksort+Insertion(k=20) | 10685 |
| Quicksort+Insertion(k=30) | 10586 |
| Quicksort+Insertion(k=50) | 10329 |
| Quicksort+Insertion(k=100) | 9134 |
| Quicksort+Insertion(k=200) | 7582 |
| Quicksort+Insertion(k=500) | 10465 |
| Quicksort+Insertion(k=1000) | 18233 |

To be more effective than just using quicksort, the value of k has to be between 50 and 500, with the most effective value measured being k=200.