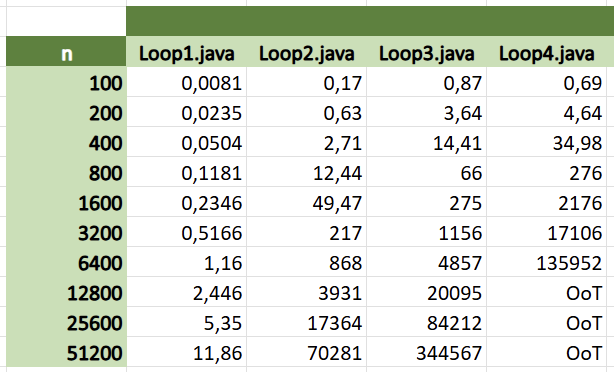
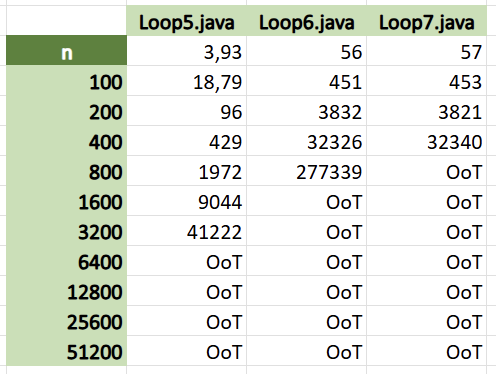
Activity 1. Loop.java 1-4



* Explain whether the different times obtained agree with what was expected, according to the theoretical complexity of the four cases.

They do. Loop1 is O(nlogn), Loop2 and Loop3 are both O(nlogn), and Loop4 is O(n).

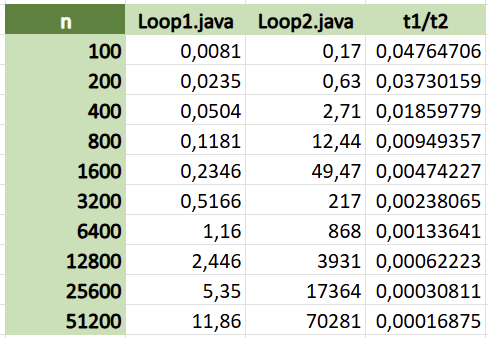
Activity 2. Loop.java 5-7



* Explain whether the different times obtained agree with what was expected, according to the theoretical complexity of the four cases.

They do. Loop5 is O(nlogn), Loop6 is O(nlogn) and Loop7 is O(n).

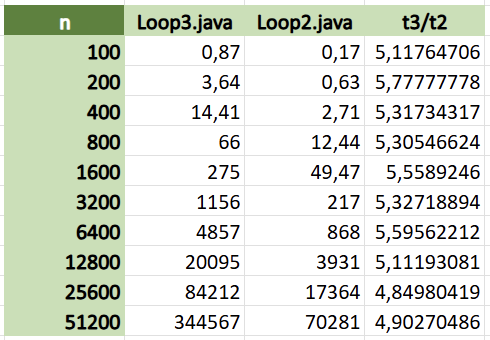
Activity 3. Two algorithms with different complexity



* Explain whether the different times and their quotient agree with what was expected according to the theoretical complexity.

Most definitely. Since Loop2 is worse than Loop1, and it’s in the denominator, t1/t2 approaches zero.

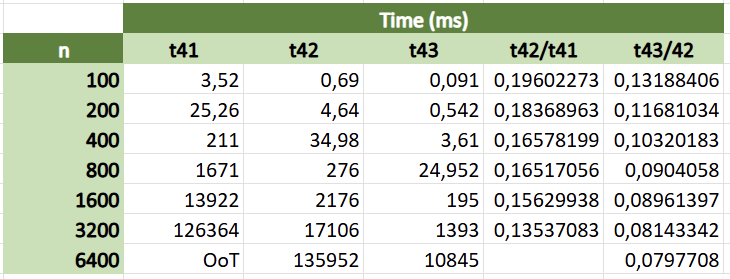
Activity 4. Two algorithms with the same complexity



* Explain whether the different times and their quotient agree with what was expected according to the theoretical complexity.

Yes, they do. Both Loop2 and Loop3 are O(nlogn), but since t3/t2 approaches an implementation constant of 5 (which is > 1), we know Loop2 is more efficient.

Activity 5. Same algorithm in different environments



* Explain whether the different times and their quotient agree with what was expected according to the theoretical complexity.

This is indeed the case; t42/t41 approaches zero because Java is faster than Python in general, a fact that is especially noticeable when executing complex algorithms, while with t43/t42 the same result is explained by the optimization in t43, which has an increasingly large effect on the execution time as the problem size gets bigger.