In the first part of this talk, we give an analysis of the sorting algorithm TimSort, which is implemented in many popular programming languages such as Python, Java,... In the second part, we present some surprising experimental results on the execution time of basic algorithms run on modern processor. We explain these observations by theoretical analysis of the algorithms taking features of modern achitecture into account.

Though quite different, both parts to try to bridge the gap between textbook algorithms and their "real life" implementations: understanding why TimSort is becoming a very popular sorting algorithm, and how we may have to change our way of analyzing algorithms considering the progresses made in computer architecture.