# BIG-O NOTATION

# BIG-O NOTATION

THIS EXPRESSES THE COMPLEXITY OF AN ALGORITHM

AN ALGORITHM WHOSE COMPLEXITY DOES NOT CHANGE WITH THE INPUT SIZE IS O(1)

THE ALGORITHM IS SAID TO HAVE CONSTANT TIME COMPLEXITY

IT TAKES THE SAME AMOUNT OF TIME EVEN IF THE INPUT SIZE IS DOUBLED, TRIPLED OR INCREASED TO ANY LEVEL

# IF "N" IS THE SIZE OF THE INPUT....

THE COMPLEXITY OF AN ALGORITHM IS O(N) IF THE TIME TAKEN BY THE ALGORITHM INCREASES LINEARLY WHEN N INCREASES

THE COMPLEXITY OF AN ALGORITHM IS $O(N^2)$ IF THE TIME TAKEN BY THE ALGORITHM INCREASES QUADRATICALLY WHEN N INCREASES

# WHAT IS THE COMPLEXITY OF COMMON OPERATIONS?

THE COMPLEXITY OF AN ALGORITHM IS O(N) IF THE TIME TAKEN BY THE ALGORITHM INCREASES LINEARLY WHEN N INCREASES

THE COMPLEXITY OF AN ALGORITHM IS $O(N^2)$ IF THE TIME TAKEN BY THE ALGORITHM INCREASES QUADRATICALLY WHEN N INCREASES

LOWER ORDER TERMS AND CONSTANTS DO NOT MATTER WHILE EXPRESSING COMPLEXITY, THE ASSUMPTION IS THAT N IS VERY LARGE

$O(N^2 + 1000)$ IS EQUIVALENT TO $O(N^2)$

$O(N^2 + N)$ IS EQUIVALENT TO $O(N^2)$

# WHICH ALGORITHMS ARE FASTER?

TIME TAKEN

$O(1) < O(N) < O(N^2) < O(N^3)$

FASTEST SLOWEST