# Algorithms Lecture 1

## RAM model –primitive data types (not strings)

-primitive operations – run time is independent of the inputs size

-length n -> f(n) =n^2 +7 (don’t care about 7) = O(n^2)

-loop run n times -> n^3 +7n (still don’t care about 7)

-O(n^2) = cn^2

-O <=

- omega - >=

theta - =

o < // w>

-T(n) = O(n^2) ok – T(n) <= O(n^2) clearer

n^2, n^2+7, n^2+n <= O(n^2)

= 2n^2

-efficient – number of ops, big O

* L – list with length 10^9 +1
  + In L, every number from 1 – 10^9 appears, one of these numbers appears twice.
  + Obvious alg- go through list / time: 10^9+1 θ (n)/ space: θ (n)
  + Better alg- add entries of L / ∑n/ i=1 (i=n(n+1)/2) /S =∑x=L (x)=(10^9(10^9+1)/2 +y)
  + Return S- x Time : θ(n)/ space: θ (1)
* O(n) = pretty much done/ = size of input/usually best can do
* Worst case analysis
  + “for all possible inputs”/ “guaranteed, no matter what”/ “Adversarial”
  + want is worse possible input to make run as long as possible
* Average-case
* Need to make upper bound as tight as possible