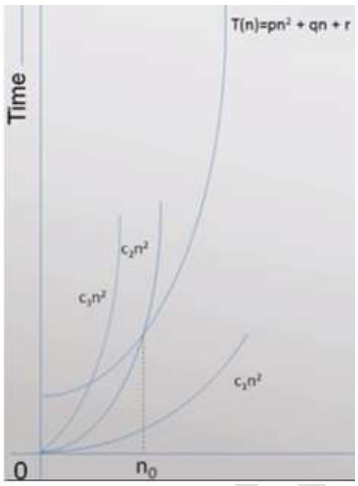
The Big-O Notation

Assume the order of time of an algorithm is a **quadratic** time as displayed in the graph. Our job is to find an **upper bond** for this function **T(n)**. Consider a function c_1n^2 ← never over take **T(n)**

C_2n^2 such that its greater than **T(n)** for **$n>n_0$** . In this case we say that C_2n^2 is an upper bond of **T(n)**

But we can come up with many functions satisfy this condition. We need to be precise.



OBSERVATIONS:

$\forall n > n_0$

$c_2n^2 \geq pn^2 + qn + r$

Big Oh **$O(n^2)$**: **f(n)**: there exist positive constants **c** and **n_0** such that **$0 \leq f(n) \leq cn^2$** for all **$n \geq n_0$**

In general

O(g(n)) : **f(n)**: there exist positive constants **c** and **n_0** such that **$0 \leq f(n) \leq cg(n)$** for all **$n \geq n_0$**

Example 1:

$5n^2 + 6 \in O(n^2)$??? ✓

Find **cn^2** ➔ c=6 and n_0=3

➔ c=5.1 n_0=8

Example 2:

$5n + 6 \in O(n^2)$??? ✓

Find **cn^2** ➔ c=11 and n_0=1

Example 3:

$n^3 + 2n^2 + 4n + 8 \in O(n^2)$??? ×

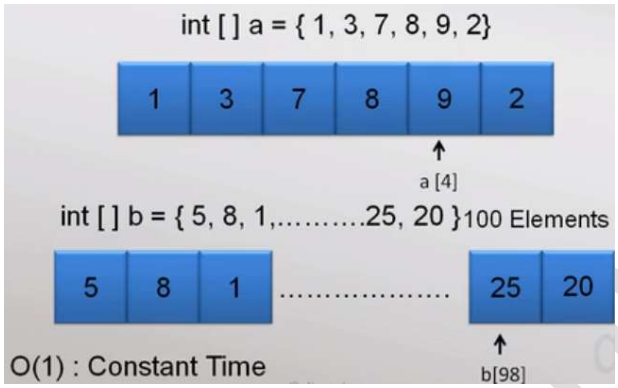
Find $cn^2 \geq n^3 + 2n^2 + 4n + 8$??? ×

$$a_m n^m + a_{m-1}n^{m-1} - - - - - - - - - - + a_0 \in O(n^m)$$

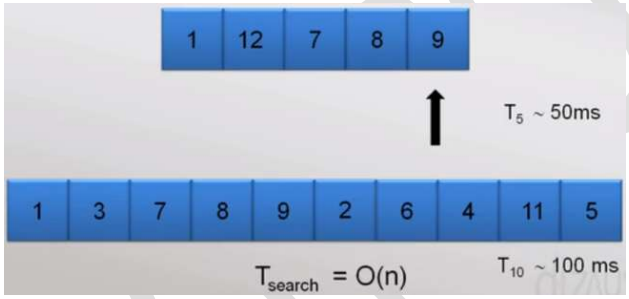
$$\log n \leq \sqrt{n} \leq n \leq n\log n \leq n^2 \leq n^3 \leq 2^n \leq n!$$

What does it mean?

Array element access:



Array element search:



Bubble sort algorithm: