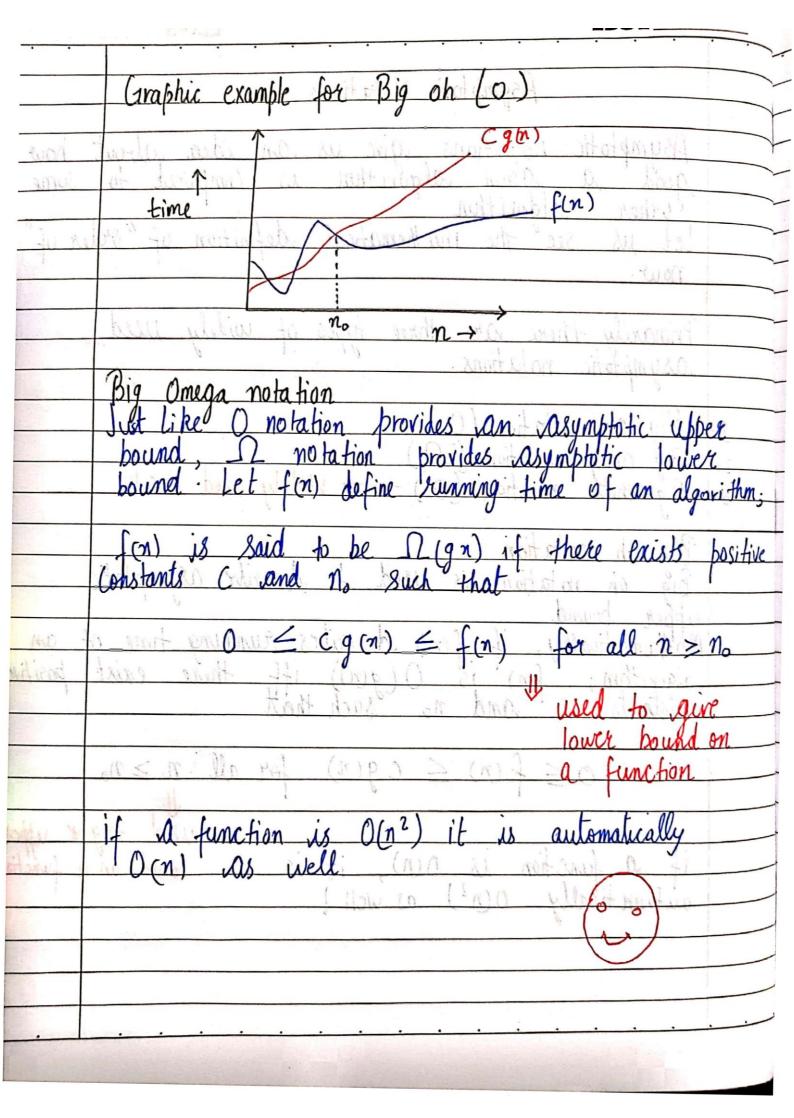
Asymptotic Notations  Asymptotic notations give us an idea about how acod a given algorithm is compared to some other algorithm.  Let us see the mathematical definition of "order of" now.  Primarily there are three types of widely used asymptotic notations.  17 Big Oh notation (O)  3. Big onega notation (O)  3. Big oh notation (O)  4. Big oh notation (O)  4. Widely used one!  Big oh notation  5. Big oh notation  6. O Asscribe asymptotic upper bound.  Mathematically, if $f(n)$ describes auroung time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants (and no such that $O \leq f(n) \leq c g(n)$ for all $n \geq n_0$ 4. If a function is $O(n)$ , it is bound on a function automatically $O(n^2)$ as well!	
Asymptotic notations give us an idea about how good a given algorithm is compared to some other algorithm  Let us see the mathematical definition of "order of" now.  Primarily there are three types of widely used asymptotic notations.  1. Big Oh notation (O)  3. Big omega notation (Q)  3. Big theta notation (Q) Widely used one!  Pig oh notation  Big oh notation  Big oh notation is used to describe asymptotic upper bound  Mathematically, if f(n) describes running time of an algorithm; f(n) is O(g(n)) iff there exist positive constants ( and no such that	Asymptotic Notations address address
Primarily there are three types of widely used asymptotic notations.  1. Big Oh notation (O)  2. Big omega notation (O)  3. Big theta notation (O) Widely used one!  Big Oh notation  Big oh notation is used to discribe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( and no such that $O \subseteq f(n) \subset Cg(n)$ for all $n > n_0$	
Primarily there are three types of widely used asymptotic notations.  1. Big Oh notation (O)  2. Big omega notation (O)  3. Big theta notation (O) Widely used one.  Big oh notation  Big oh notation is used to discribe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( and no such that $O \subseteq f(n) \subset Cg(n)$ for all $n > n_0$	Asymptotic notations sive us an idea about how
Primarily there are three types of widely used asymptotic notations.  1. Big Oh notation (O)  2. Big omega notation (O)  3. Big theta notation (O) Widely used one.  Big oh notation  Big oh notation is used to discribe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( and no such that $O \subseteq f(n) \subset Cg(n)$ for all $n > n_0$	good a given algorithm is compared to some
Primarily there are three types of widely used asymptotic notations.  1. Big Oh notation (O)  2. Big omega notation (O)  3. Big theta notation (O) Widely used one!  Big Oh notation  Big oh notation is used to discribe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( and no such that $O \subseteq f(n) \subset Cg(n)$ for all $n > n_0$	other algorithm
Primarily there are three types of widely used asymptotic notations.  1. Big Oh notation (O)  2. Big omega notation (O)  3. Big theta notation (O) Widely used one!  Big Oh notation  Big oh notation is used to discribe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( and no such that $O \subseteq f(n) \subset Cg(n)$ for all $n > n_0$	Let us see the mathematical definition of "order of"
1. Big oh notation (0) 2. Big omega notation ( $\Omega$ ) 3. Big theta notation ( $\Omega$ ) Widely used one.  Big oh notation is used to describe asymptotic upper bound.  Mathematically, if $f(n)$ describes summing time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( $O$ ) and no such that	now.
1. Big oh notation (0) 2. Big omega notation ( $\Omega$ ) 3. Big theta notation ( $\Omega$ ) Widely used one.  Big oh notation is used to describe asymptotic upper bound.  Mathematically, if $f(n)$ describes summing time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( $O$ ) and no such that	
1. Big oh notation (0) 2. Big omega notation ( $\Omega$ ) 3. Big theta notation ( $\Omega$ ) Widely used one.  Big oh notation is used to describe asymptotic upper bound.  Mathematically, if $f(n)$ describes summing time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( $O$ ) and no such that	Primarily there are three types of widely used
1. Big oh notation (0) 2. Big omega notation ( $\Omega$ ) 3. Big theta notation ( $\Omega$ ) Widely used one.  Big oh notation is used to describe asymptotic upper bound.  Mathematically, if $f(n)$ describes summing time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( $O$ ) and no such that	asymptotic notations.
Big oh notation is used to describe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants $C$ and no such that $O \leq f(n) \leq Cg(n)$ for all $n \geq n_0$	Me Closella moter todas
Big oh notation is used to describe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants $C$ and no such that $O \leq f(n) \leq Cg(n)$ for all $n \geq n_0$	17 Big Oh notation (Oa) and noted on and the
Big oh notation is used to describe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants $C$ and no such that $O \leq f(n) \leq Cg(n)$ for all $n \geq n_0$	27 Big Omega notation (S2)
Big oh notation is used to describe asymptotic upper bound.  Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants $C$ and no such that $O \leq f(n) \leq Cg(n)$ for all $n \geq n_0$	37 Big theta notation (O) -> Widely used one!
Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( and no such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$	
Mathematically, if $f(n)$ describes running time of an algorithm; $f(n)$ is $O(g(n))$ iff there exist positive constants ( and no such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$	big in notation is a line of the second of t
$0 \le f(n) \le cg(n) \text{ for all } n \ge n_0$	Big of notation is used to asserve asymptotic
$0 \le f(n) \le cg(n) \text{ for all } n \ge n_0$	where downa
$0 \le f(n) \le cg(n) \text{ for all } n \ge n_0$	rivathematically, if f(n) alsours running time of an
$0 \le f(n) \le cg(n) \text{ for all } n \ge n_0$	algorithm; fin) is O (g(n)) Iff there exist positive
$0 \le f(n) \le cg(n) \text{ for all } n \ge n_0$	Constants C AVILLA TO SUCH THAN
and a single for the second se	Com ( cold) loy all m > m
if a function is $O(n)$ , it is bound on a function automatically $O(n^2)$ as well!	0 = 7 (n) = c g c s) pot aux n = no
if a function is $O(n)$ , it is bound on a function automatically $O(n^2)$ as well!	when his in the mission of the it is there
automatically $O(n^2)$ as well!	if a function is $n(n)$ it is bound on a sunction
autilians out a work.	a formatically $O(n^2)$ as yiell 1
	au min in any
	Li / E M Line Man do die Man



	Graphic es	cample for Big omega (D)	
		f(n)	
	<b>^</b>	(N) 1 4 1 7 1 7 1 7 1 7 1 7 1 7 1 7 1 7 1 7	
_	time	C9(n)	
	(10) 6	A	
			K
		<b>—</b>	- 14
_		$n_0$ $n \rightarrow$	
	Big the ta	rotation s we of seeds to ear hide	
. 1	1 1 1		
	let for a	efine running time of an algorithm	
[Y][]	May sat 10	12A(1) (10A1) (10A(1)	
1	(m) 18 80	id to be O(gn) iff fon) is O(gn) and	
	'f(n) is	(12(g(n)) & with while state in	
5	1		
	Mathematica	ly n+ n tost grove in + next saint	
NV:	Using Yester	in A has (r) ()	
	0 4	$f(n) \leq C_1 g(n) + n > n_0 - Sufficiently large value$	y
_		= $(-9(n)) \leq (f(n)) + n \geq n_0 > 0 \leq n$	
_	0 =	$\leq C_2 g(n) \leq f(n) + n \geq n_0 > n_0$	
-	Managar ha	the the equations, we get:	
	morgany 100		
7.		$0 \leq C_2 g(n) \leq f(n) \leq C_1 g(n) + n \geq n_0$	<b>—</b>

The equation simply means there exist positive constants and C2 Such that f(n) is sandwiched between C2 g(n) and C, g(n)