BigOh

Thursday, July 29, 2021 4:20 PM

$\log_2 16 = 4 \rightarrow \log_b a = y$

$\boxed{a = 2^y} \Rightarrow 16 = 2^4$

$n > \log_2(n)$

$\checkmark\ 16 > \log_2(16)$
4

$\rightarrow \boxed{b^y = a}$

$2^y = 16$

$2^4 = 16$

$\checkmark\ 128 > \log_2(128)$
6

$\rightarrow 2^y = 128 \rightarrow$

$\log_2(1) \rightarrow 2^y = 1 \rightarrow 2^0 \Rightarrow 1$

$\log_8(1) \rightarrow 8^y = 1 \rightarrow$

$32 > \log_2(30) \approx 5$

$\longrightarrow 2^y = 30 \rightarrow 32$

$\log_2 2 \rightarrow y = \log_b a \rightarrow \boxed{b^y = a}$

$\boxed{2^y = 2} \rightarrow y = 1$

$\log_8 8 =$

$8^y = 8$

$\boxed{\log_a a = 1}$

$$\boxed{\log_b 1 = 0}$$

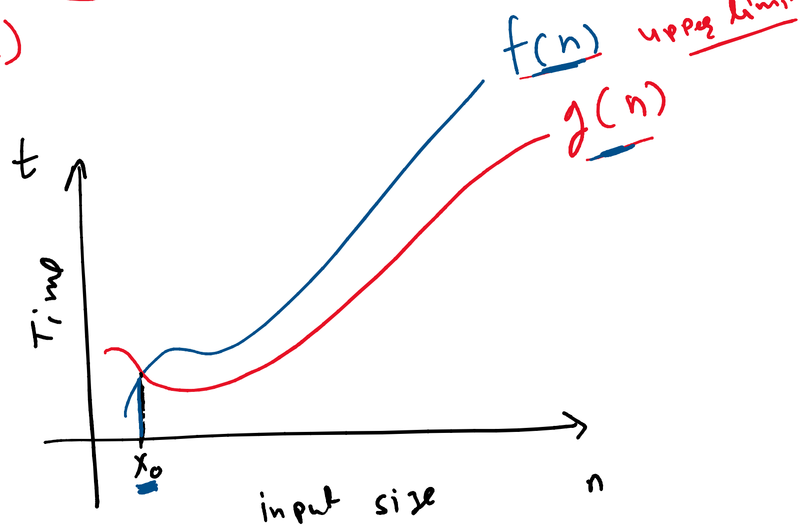
$$\boxed{\log_c (a.b) = \log_c a + \log_c b}$$

$$\boxed{\log_b a = \log_e a / \log_e b}$$

Big Oh (worst case)

$T(n) = O(n) + O(\log(n))$

$T(n) = f(n) + g(n)$



$$\boxed{g(n) \le c.f(n)}$$

where $x > x_0$ & $c \ge 0,\ n_0 \ge 1$

$$\boxed{g(n) = O(f(n))}$$

$$O(5n + n^2 + c) \rightarrow O(n^2)$$

g(n) f(n)

$$\boxed{g(n) \le c.f(n)}$$

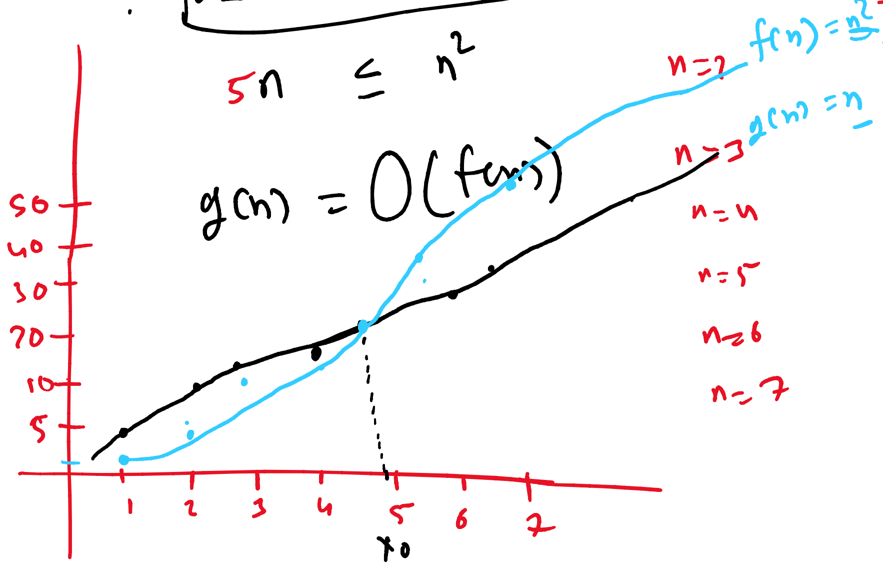
$n=1$ $5n$ n^2

I 1

$$\boxed{g(n) \le c \cdot f(n)}$$

$$5n \le n^2$$

$$g(n) = O(f(n))$$



	$5n$	n^2
$n=1$	5	1
$n=2$	10	4
$n=3$	15	9
$n=4$	20	16
$n=5$	25	25
$n=6$	30	36
$n=7$	35	49
	⋮	⋮

$$T(n) = O(c \cdot n) + O(n^2) + c$$

$$\boxed{T(n) = O(n^2)}$$

```
for (j=1; j<m; j++)        2m
{
  k=j+j;      # m
  printf(k);  # m
}
for (i=1; i<n; i++)
{
  print(i);   # n
}
```

j =	1	2	3	4	.....	k
iteration	1	2	3	4	.....	m

When $j = k$ condition False

$\boxed{k < m} \rightarrow \boxed{k = m}$

$O(\quad)$

$$T(n) = O(m + n)$$

```
for (i=1; i<=n; i+5)
{
    print(i)  # K
}
```

steps / iterations	1	2	3	4 ...	K
i	1	6	11	16 ...	1+5*( )
	1+5*0	1+5*1	1+5*2	1+5*3	

$$\rightarrow \boxed{1 + 5 * (k-1)}$$

$$\boxed{i = 1 + 5 * (k-1)}$$

$$i = n$$

$$1 + 5 * (k-1) = n$$

$$5 * (k-1) = n-1$$

$$5k - 5 = n-1$$

$$5k = n-4$$

$$\boxed{K = \frac{n-4}{5}}$$

$$T(n) = \frac{n-4}{5} \Rightarrow \boxed{\frac{n}{5}} - \cancel{\frac{4}{5}}$$

const

$$= \frac{n}{5} \Rightarrow O(n)$$

$$\boxed{T(n) = O(n)}$$

$$\boxed{T(n) = O(n)}$$

$f(n) = \frac{n}{5}$

$f(n) = O(g(n))$

$g(n) = n$

$f(n) \le c \cdot g(n)$ $\quad n > n_0, \quad c, n_0 \in R \quad c>0, n_0>=1$

$$\boxed{f(n) = O\, g(n)}$$

$f(n)$	$g(n)$
$\frac{n}{5}$	n
1	5
2	10
...	...

$n=5$

$n=10$

$c=1$

$f(n) \le g(n)$

$f(n) = O(g(n))$

(2)

```
i = n
WHILE  [i >= 1]
DO
    print . i    # k+1
    [i = i/2]
DONE
```

$\left(\frac{n}{4}\right)^{\frac{1}{2}}$

$T(n) = k+1$

Time complexity 2.

```
for (i = n; i >= 1; i = i/2)
{     print (i);
}
```

python

```
i = n
while i >= 1:
    print (i)
    i = i/2
```

$i = i/2$

Iteration	1	2	3	4	5	...	k	k+1
i	n	n/2	n/4	n/16	n/16	...	2.	1.
i	$\frac{n}{2^0}$	$\frac{n}{2^1}$	$\frac{n}{2^2}$	$\frac{n}{2^3}$	$\frac{n}{2^4}$	... 2^5	$\frac{n}{2^{k-1}}$	$\frac{n}{2^k}$

$i = 1$

$\boxed{\frac{n}{2^k} = 1}$

$n = 2^k$

$\log_b a = y$

$y = b^1$

$\Rightarrow 2^k = n$

$\Rightarrow \log_2(2^k) = \log_2(n)$

$\Rightarrow k \log_2(2) = \log_2(n)$

$\Rightarrow \boxed{k = \log_2(n)}$

$\boxed{\log m^n = n \log(m)}$

$T(n) = \log_2(n) + 1$

$$\boxed{T(n) = O(\log_2(n))}$$

③

```
for (i = 1; i <= n; i = i * 2)
{
    print (i); # k
}
```

}

iteration	1	2	3	4	5	...	k	k+1
i	$1=2^0$	$2=2^1$	$4=2^2$	$8=2^3$	$16=2^4$	...	2^{k-1}	2^k

$i = k+1 \to$ condn false

$\boxed{i = n}$

$i \Rightarrow 2^k \Rightarrow \boxed{2^k = n}$

$\log 2^k = \log(n)$

$\boxed{k = \log(n)}$

$\boxed{T(n) = O(\log(n))}$

```
for ( i=1 ; i<n ; i = i+5)
    { print (i)
    }
for ( j= n ; j >1 ; j = j-8)
    { print (j)
    }
```

} $O(n)$

```
for ( j =1 ; j<=n ; j= j*5)
    { print (j) ; }
```

$\to O(\log_5(n))$

```
for ( j= n; j>=1; j=j/3)
    { print ( j); }
```

$\to O(\log_3(n))$

④

for (i=2 ; i<=n ; i = i^2)
{ print (i) ; # k
}

Iteration	1	2	3	4	... 5	k	k+1
i	2	4	16	256	65536 ...		2^{2^k}
	2^1	2^2	2^4	2^8	2^{16}		
	2^{2^0}	2^{2^1}	2^{2^2}	2^{2^3}	2^{2^4}	$2^{2^{k-1}}$	

for $i = 2^{2^k}$ condition becomes false

$i = n$

$2^{2^k} = n \Rightarrow 2^{(2^k)} = n$

$\log(2^{(2^k)}) = \log(n) \Rightarrow 2^k \log(2) = \log(n)$

$2^k = \log(n)$

$\log(2^k) = \log(\log(n))$

$k \cdot \log(2) = \log(\log(n))$

$k \cdot \log(2) = \ldots$

$$\boxed{k = \log(\log(n))}$$

$$\boxed{T(n) = O(\log \log n)}$$

---

(5) `for ( i = n; i >= 2; i = √i )`
```
{   print(i);
}
```

$\rightarrow \sqrt{i} = (i^{1/2})$

$(n^{1/2})^{1/2} \rightarrow n^{1/4}$

$(n^{1/4})^{1/2} \rightarrow n^{1/8}$

Iteration	1	2	3	4	5	...	k	k+1
i	$n^{\frac{1}{2^0}}$	$n^{\frac{1}{2^1}}$	$n^{\frac{1}{2^2}}$	$n^{\frac{1}{2^3}}$	$n^{\frac{1}{2^4}}$	...	$n^{\frac{1}{2^{k-1}}}$	$n^{\frac{1}{2^k}}$

for $i = n^{\frac{1}{2^k}}$ condition will false

$i = 2 \Rightarrow n^{\frac{1}{2^k}} = 2 \Rightarrow \log_2\left(n^{\left(\frac{1}{2^k}\right)}\right) = \log_2(2)$

$\frac{1}{2^k} \cdot \log(n) = 1 \Rightarrow 2^k = \log(n) \Rightarrow \log(2^k) = \log(\log(n))$

$$\boxed{K = \log(\log(n))}$$

$$\boxed{K = \log(\log(n))}$$

$$\boxed{T(n) = O(\log \log n)}$$