Análisis de Algoritmos
TP 2

Equipo: Tania Patiño
Javier Sagastuy
Ernesto Valdés
Víctor Peña

Análisis de Algoritmos

- ¿Cuál de las siguientes expresiones es $\theta(n^2)$?, encontrar c_1, c_2, n.

- a) $60\ n^2 + 5n + 1$
- b) $3n^2 + 2\ n\ \lg n$
- c) $2+4+6+...+2n$
- d)
```
for i = 1 to n
    for j = 1 to i
        for k =1 to i
            x = x+1
```

a) $60\, n^2 + 5n + 1$

Demostración:

$$60n^2 + 5n + 1$$

$$c_1 n^2 < 60n^2 + 5n + 1 < c_2 n^2$$

$$c_1 < 60 + \frac{1}{n} + \frac{1}{n^2} < c_2$$

Sabemos que la n más pequeña para que la expresión tenga sentido es n=1 por lo que:

$$c_1 < 60 + 1 + 1 < c_2$$

$$c_1 < 62 < c_2$$

$$\therefore c_2 = 62$$

a) 60 n^2 + 5n +1 (continuación)

Ahora si hacemos tender a n a infinito observamos que los factores $\frac{1}{n}$ y $\frac{1}{n^2}$ se vuelven cero, por lo que:

$$c_1 < 60 < c_2$$

$$\therefore c_1 = 60$$

En conclusión $60n^2$+5n +1 queda perfectamente acotada con $c_1 = 60$ y $c_2 = 62$ demostrando que la función es de orden $\theta(n^2)$.

b) $3n^2 + 2n \lg n$

Demostración: Buscamos constantes positivas c_1 y c_2 y un natural n tal que

$$c_1 n^2 \leq 3n^2 + 2n \lg n \leq c_2 n^2$$

esto es,

$$c_1 \leq 3 + \frac{2 \lg n}{n} \leq c_2$$

Si $$n \geq 1 \implies \frac{2 \lg n}{n} \geq 0 \implies 3 + \frac{2 \lg n}{n} \geq 3 = c_1$$

b) $3n^2 + 2n \lg n$ (continuación)

Analicemos e intentemos acotar superiormente la siguiente función:

$$f(n) = \frac{\lg n}{n} = \frac{1}{\ln 2} \cdot \frac{\ln n}{n}$$

Derivando:

$$f'(n) = \frac{1}{\ln 2} \cdot \left(\frac{\frac{1}{n}(n) - (1)(\ln n)}{n^2} \right) = \frac{1}{\ln 2} \cdot \left(\frac{1 - \ln n}{n^2} \right)$$

Buscamos los puntos críticos:

$$f'(n) = 0 \iff 1 - \ln n = 0 \iff \ln n = 1 \iff n = e$$

b) $3n^2 + 2\, n\, lgn$ (continuación)

Derivamos nuevamente para obtener:

$$f''(n) = \frac{1}{\ln 2} \cdot \left(\frac{-\frac{1}{n}(n^2) - (1 - \ln n)(2n)}{n^4} \right)$$

Evaluamos en el punto crítico:

$$f''(e) = \frac{1}{\ln 2} \cdot \left(\frac{-\frac{1}{e}(e^2) - (1 - \ln e)(2e)}{e^4} \right) = \frac{1}{\ln 2} \cdot \left(\frac{-e - 0}{e^4} \right) < 0$$

Por lo tanto, en n=e tenemos un mínimo local, pero como:

$$\lim_{n \to \infty} \frac{\ln n}{n} = 0$$

Entonces el mínimo es global.

b) $3n^2$ + 2 n lgn (continuación)

De este modo,

$$f(n) = \frac{\lg n}{n} \leq \frac{1}{e \ln 2}$$

esto implica que

$$3 + \frac{2 \lg n}{n} \leq 3 + \frac{2}{e \ln 2} \leq 3 + 2 = 5 = c_2$$

Finalmente:

$$3 \leq 3 + \frac{2 \lg n}{n} \leq 5 \implies 3n^2 \leq 3n^2 + 2n \lg n \leq 5n^2 \quad \forall n \geq 1$$

$$\therefore\ 3n^2 + 2n \lg n \in \Theta(n^2)$$

c) 2+4+6+...+2n

- Buscar c_1, c_2 y n que cumplan con $\theta(n^2)$.

Demostración: Tenemos que $C_1 <= 2+4+6+...+2n <= C_{2}$,

$C1 * n^2 <= 2+4+6+...+2n <= C2 * n^2$

Por inducción:

$2 (1 + 2 + 3 + ... + n) = 2 (n (n+1)) / 2$

$= n (n+1)$

$= n^2 + n$

$C2 * n^2 <= n^2 + n <= C1 * n^2$

C2 <= 1 + 1/n <= C1

c) 2+4+6+...+2n (continuación)

Entonces

$n >= 1$ implica que $1 /n <= 1$

- Para C_1 tenemos que

  $C_1 * 1 + 1/n <= 1+1 = 2$

  $C_1 = 2$

- Para C_2 tenemos que

$C_2 = 1$ $\therefore$ $\theta(n^2)$

```
d)  for i = 1 to n
        for j = 1 to i
            for k =1 to i
                x = x+1
```

Solución:

1) Analizar cuántas veces se ejecuta x = x+1

```
for i = 1 to n          ──────►  n veces
    for j = 1 to i      ──────►  i veces
        for k =1 to i   ──────►  i veces
            x = x+1
```

Por lo tanto, x=x+1 se ejecuta:

$$\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$$

2) Tomando la ecuación se probará que no es cuadrática:

$c1n^2 <= n(n+1)(2n+1)/6 <= c2n^2$

Si desarrollamos llegamos a:

$c1 <= (2n + 3 + 1/n)/6 <= c2$ Por lo tanto no es una cuadrática

3) Se probará que es cúbica:

$c1n^3 <= (2n^3 + 3n^2 + n)/6 <= c2n^3$

$c1 <= (2 + 3/n + 1/n^2)/6 <= c2$

Si n tiende a infinito:

$c1 <= 2/6 <= c2$

Por lo tanto c1=2/6 y c2=1

Entonces es una función cúbica

Logaritmo iterado

Definimos el logaritmo iterado como sigue:

$$\lg^*(n) = \begin{cases} 0 & n \leq 1 \\ 1 + \lg^*(\lg(n)) & n > 1 \end{cases}$$

O alternativamente:

$$\lg^*(n) = \min\{i \geq 0 \mid \lg^{(i)}(n) \leq 1\}$$

Lista de funciones: Una posible manera de ordenar las funciones donde $g_i \in O(g_j)\ \forall i < j$

$g_1(n) = 1$

$g_2(n) = n^{1/\lg(n)} = 2$

$g_3(n) = \lg(\lg^*(n))$

$g_4(n) = \lg^*(\lg(n))$

$g_5(n) = \lg^*(n)$

$g_6(n) = 2^{\lg^*(n)}$

$g_7(n) = \ln(\ln(n))$

$g_8(n) = \sqrt{\lg(n)}$

$g_9(n) = \lg(n)$

$g_{10}(n) = \lg^2(n)$

$g_{11}(n) = 2^{\sqrt{2\lg(n)}}$

$g_{12}(n) = \sqrt{2}^{\lg(n)} = \sqrt{n}$

$g_{13}(n) = 2^{\lg(n)} = n$

$g_{14}(n) = \lg(n!)$

$g_{15}(n) = n\lg(n)$

$g_{16}(n) = 4^{\lg(n)} = n^2$

$g_{17}(n) = n^3$

$g_{18}(n) = (\lg(n))!$

$g_{19}(n) = (\lg(n))^{\lg(n)} = n^{\lg(\lg n))}$

$g_{20}(n) = \left(\frac{3}{2}\right)^n$

$g_{21}(n) = 2^n$

$g_{22}(n) = e^n$

$g_{23}(n) = n!$

$g_{24}(n) = (n+1)!$

$g_{25}(n) = n \cdot 2^n$

$g_{26}(n) = 2^{2^n}$

$g_{27}(n) = 2^{2^{n+1}}$

Análisis de Algoritmos

Clasificación de las funciones por orden de crecimiento

• O(1)	• O(lg * n)	• O(lg lg n)	• O(lg n)	• O(n ^{k})
1	lg * n	ln (ln n)	ln n	$\sqrt{2}^{\lg n}$
$n^{1/\lg n}$	lg *(lg n)	$2^{\lg^* n}$	$\sqrt{(\lg n)}$	$\lg^2 n$
	lg(lg* n)			$2^{\sqrt{(2\lg n)}}$

Análisis de Algoritmos

Clasificación de las funciones por orden de crecimiento

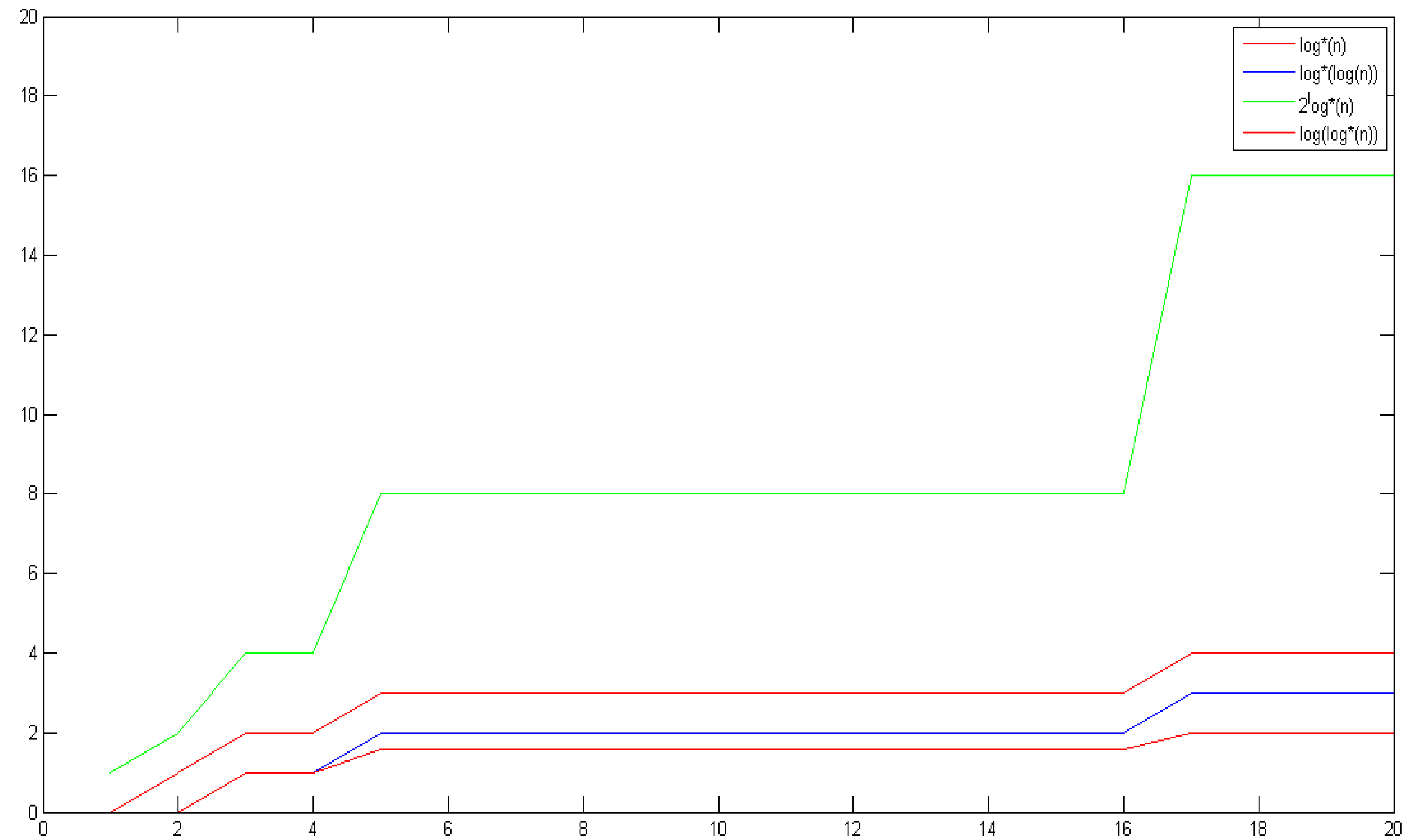
• $O(n)$	• $O(n \lg n)$	• $O(n^k)$, $_{k>1}$	• $O(c^n)$	• $O(n!)$
n	$n \log n$	n^2 - $\theta(n^2)$	e^n	$(n+1)!$
$2^{\lg n}$	$\lg^2 n$	n^3 - $\theta(n^3)$	2^n	$n!$
	$\lg (n!)$	$4^{\lg n}$ - $\theta(n^2)$	$(3/2)^n$	
			$(\lg n)!$	
			$(\lg n)^{\lg n} = n^{\lg(\lg n)}$	

Análisis de Algoritmos

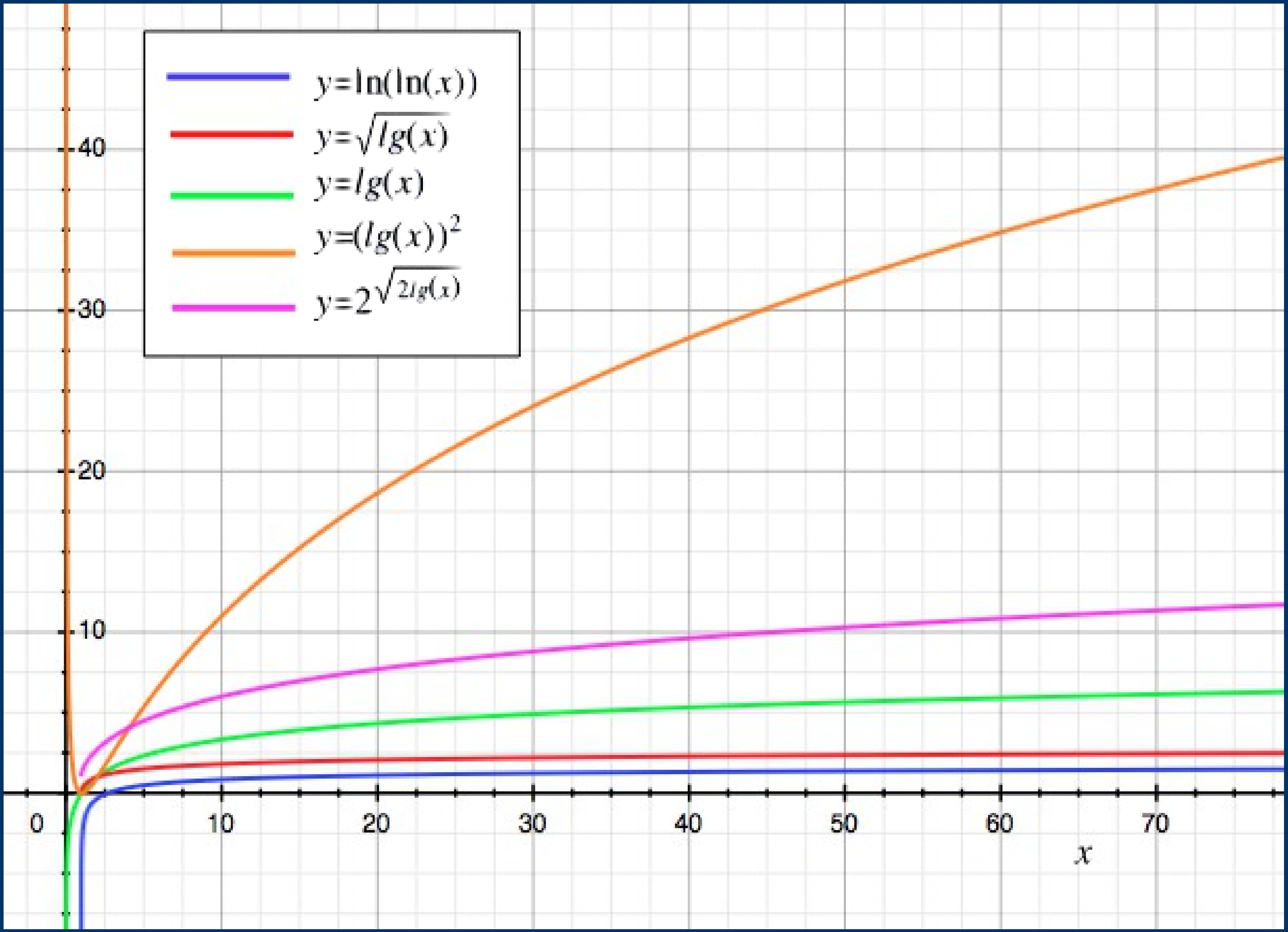
Clasificación de las funciones por orden de crecimiento

• O(n^n)	• O(2^{2^n})			
$n\,2^n$	2^{2^n}			
	$2^{2^{(n+1)}}$			

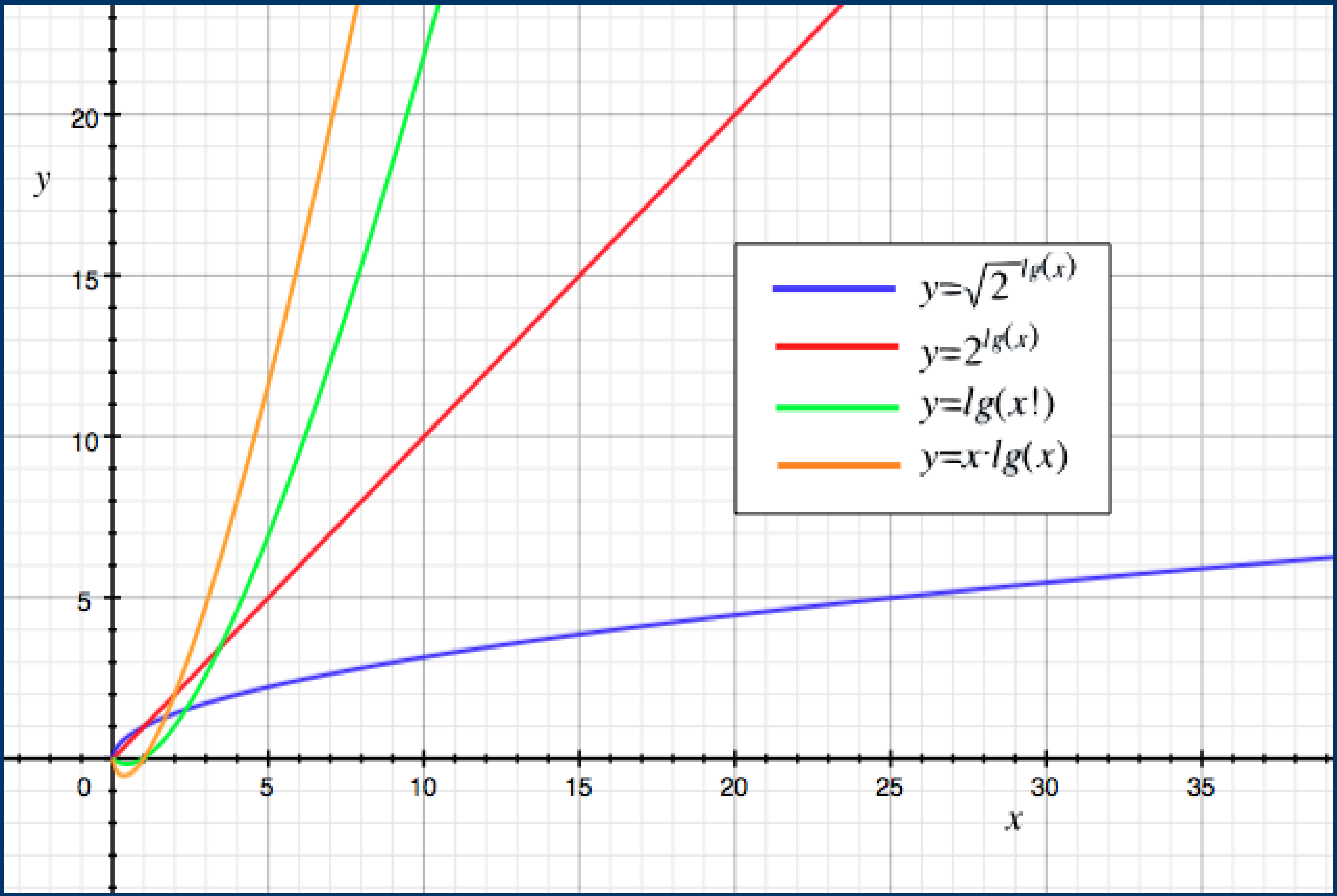
Gráficas de algunas funciones



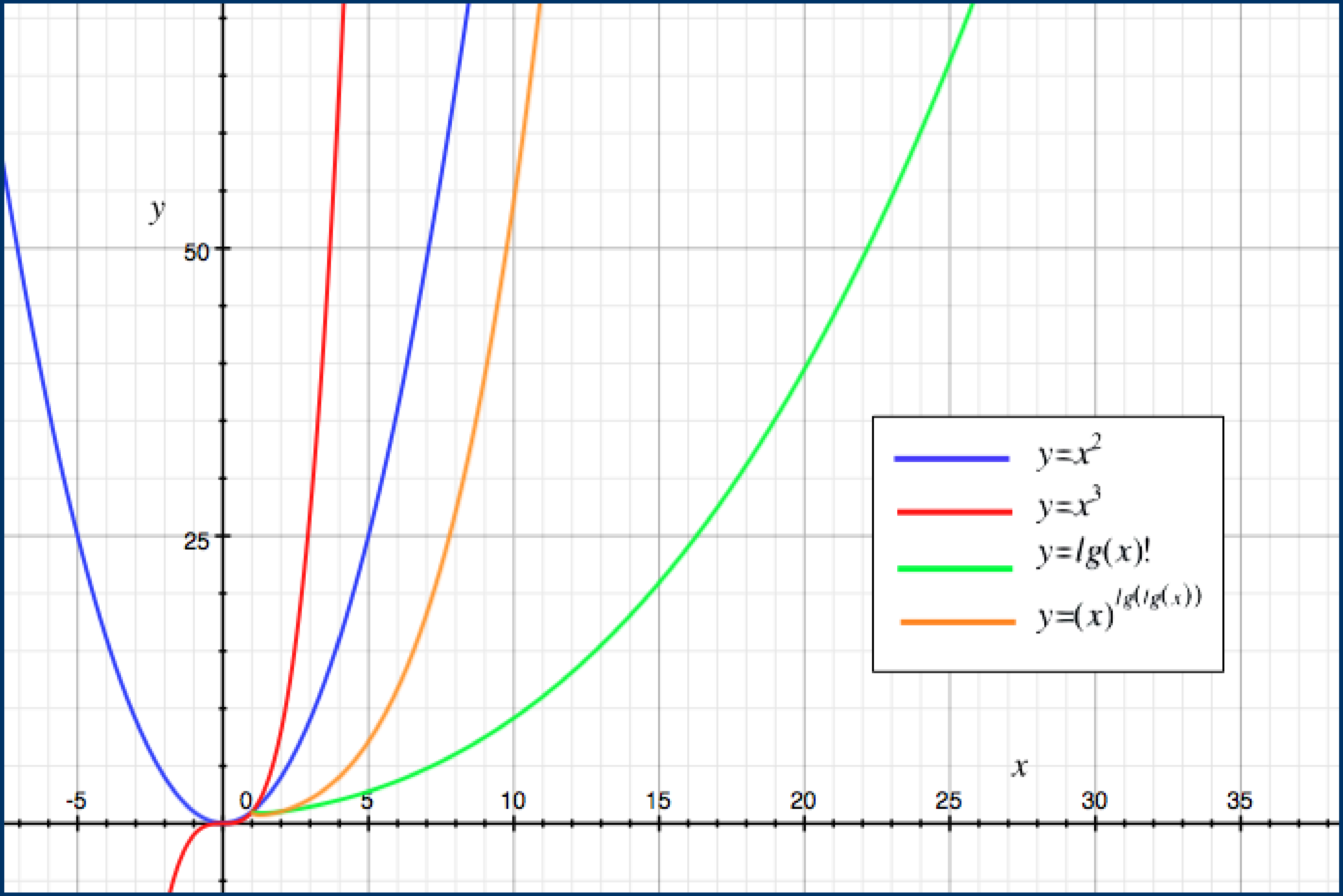
Gráficas de algunas funciones



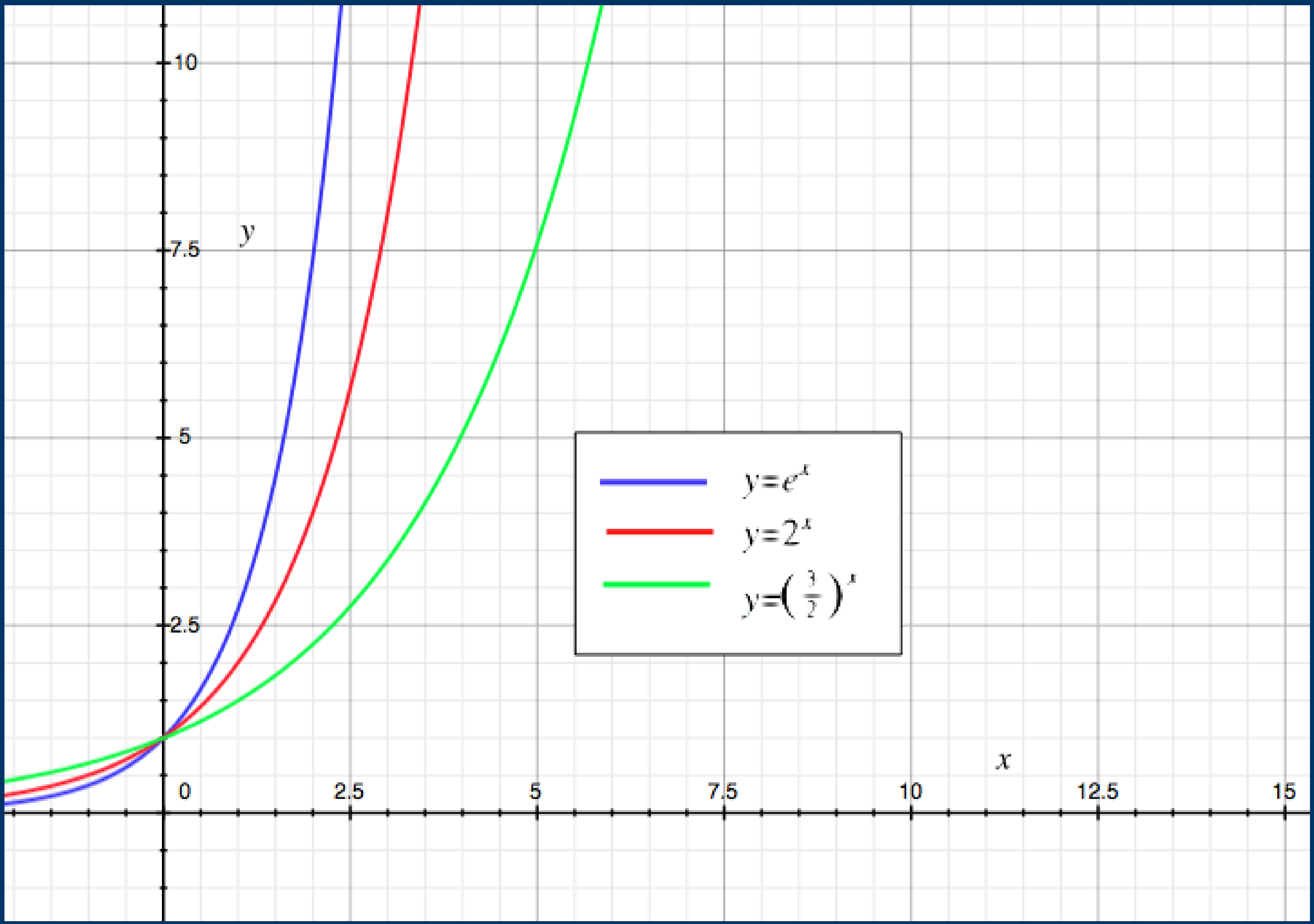
Gráficas de algunas funciones



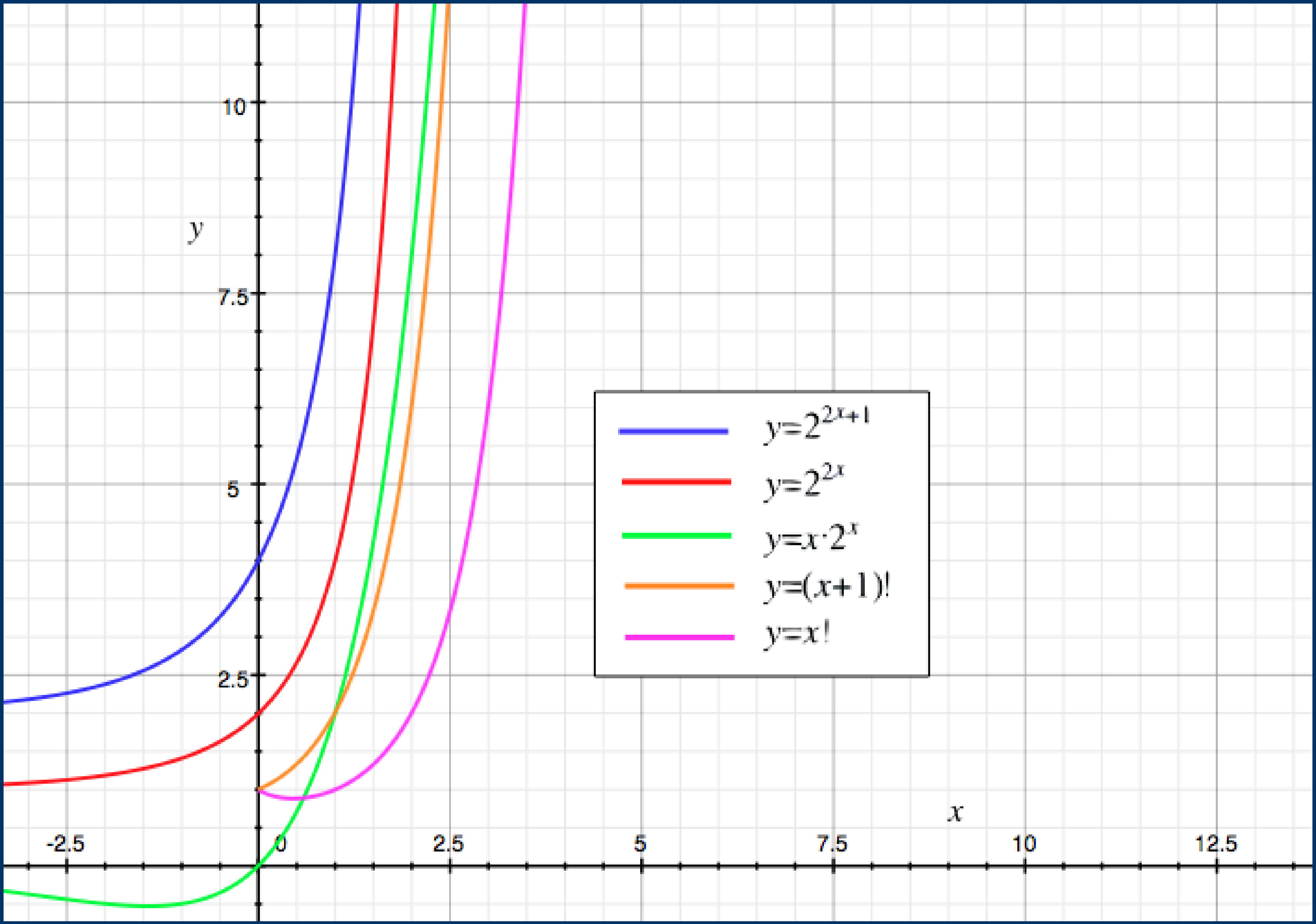
Gráficas de algunas funciones



Gráficas de algunas funciones



Gráficas de algunas funciones



Gráficas de algunas funciones