



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



-----**ANÁLISIS DE ALGORITMOS**-----

ACTIVIDAD

Graficas De Ordenes De Complejidad

PROFESOR:

Franco Martínez Edgardo Adrián

ALUMNO:

Meza Vargas Brandon David – 2020630288

GRUPO:

3CM13



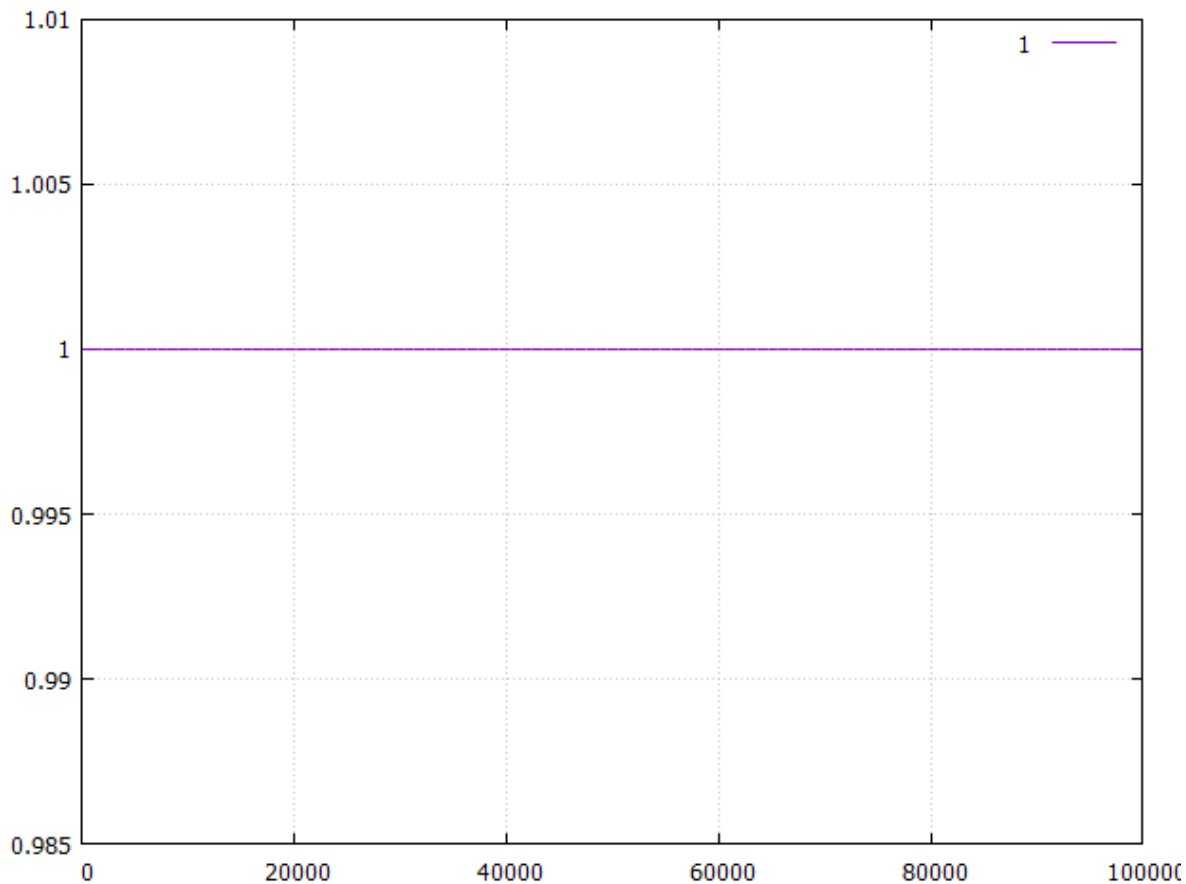
Índice

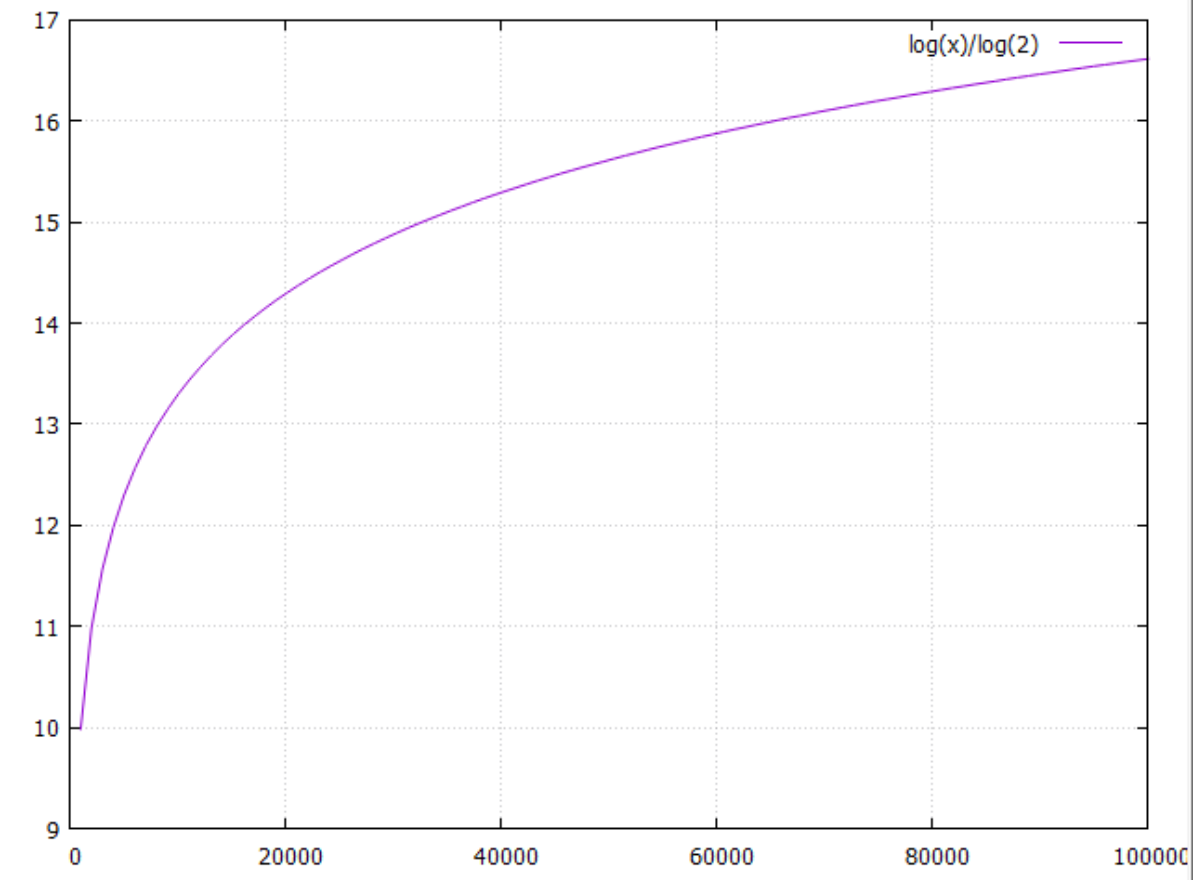
Graficas en rango $0 < n < 100,000$	4
$O(1)$ Complejidad constante	4
$O(\log(n))$ Complejidad logarítmica	5
$O(n)$ Complejidad lineal	6
$O(n\log(n))$ Complejidad “n log n”	7
$O(n^2)$ Complejidad cuadrática	8
$O(n^3)$ Complejidad cubica	9
$O(c^n); c > 1$ Complejidad exponencial	10
$O(n!)$ Complejidad factorial	10
Graficas por pares en rango $0 < n < 100$	11
$O(1)$ vs $O(\log(n))$	11
$O(1)$ vs $O(n)$	12
$O(1)$ vs $O(n\log(n))$	13
$O(1)$ vs $O(n^2)$	14
$O(1)$ vs $O(n^3)$	15
$O(1)$ vs $O(c^n)$	16
$O(1)$ vs $O(n!)$	17
$O(\log(n))$ vs $O(n)$	18
$O(\log(n))$ vs $O(n\log(n))$	19
$O(\log(n))$ vs $O(n^2)$	20
$O(\log(n))$ vs $O(n^3)$	21
$O(\log(n))$ vs $O(c^n)$	22
$O(\log(n))$ vs $O(n!)$	23
$O(n)$ vs $O(n\log(n))$	24
$O(n)$ vs $O(n^2)$	25
$O(n)$ vs $O(n^3)$	26
$O(n)$ vs $O(c^n)$	27
$O(n)$ vs $O(n!)$	28
$O(n\log(n))$ vs $O(n^2)$	29
$O(n\log(n))$ vs $O(n^3)$	30

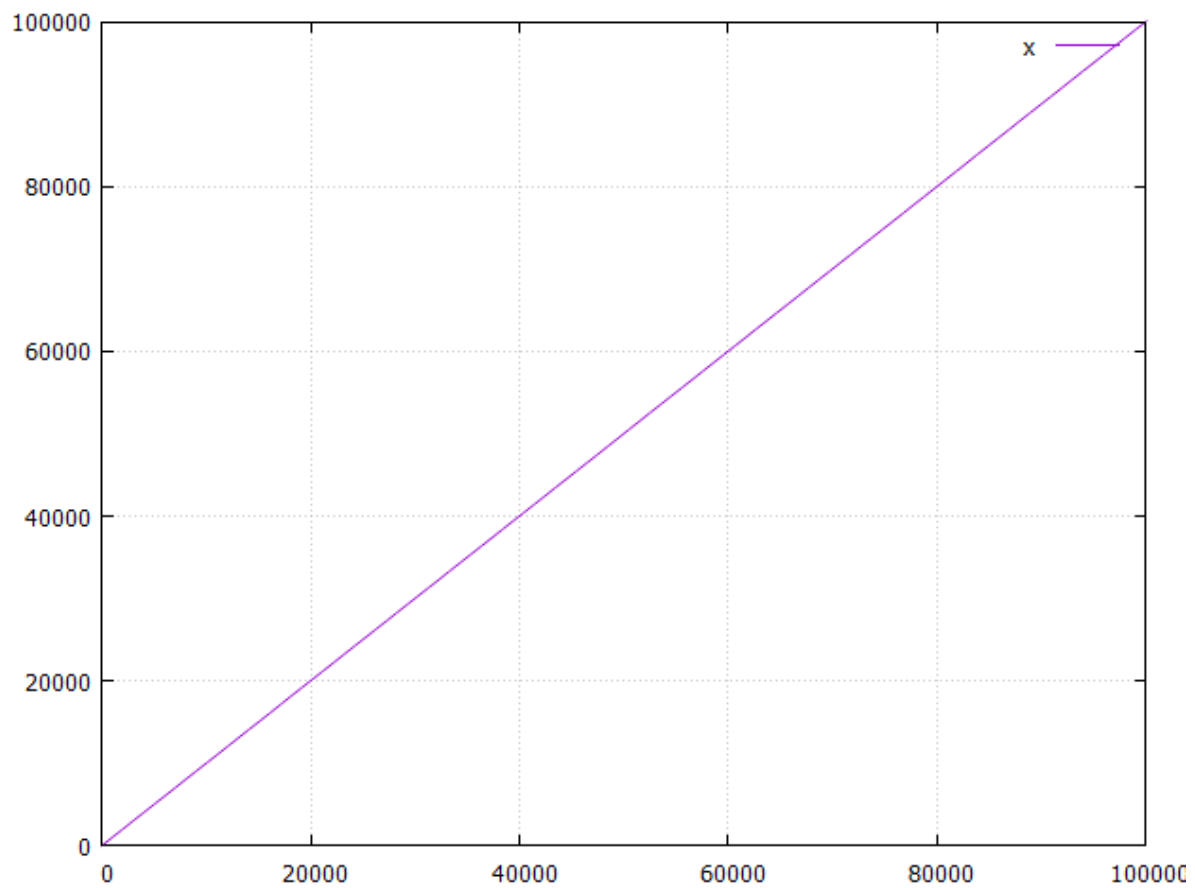
$O(n \log(n))$ vs $O(c^n)$	31
$O(n \log(n))$ vs $O(n!)$	32
$O(n^2)$ vs $O(n^3)$	33
$O(n^2)$ vs $O(c^n)$	34
$O(n^2)$ vs $O(n!)$	35
$O(n^3)$ vs $O(c^n)$	36
$O(n^3)$ vs $O(n!)$	37
$O(c^n)$ vs $O(n!)$	38
Grafica comparativa de todos los órdenes de complejidad	39

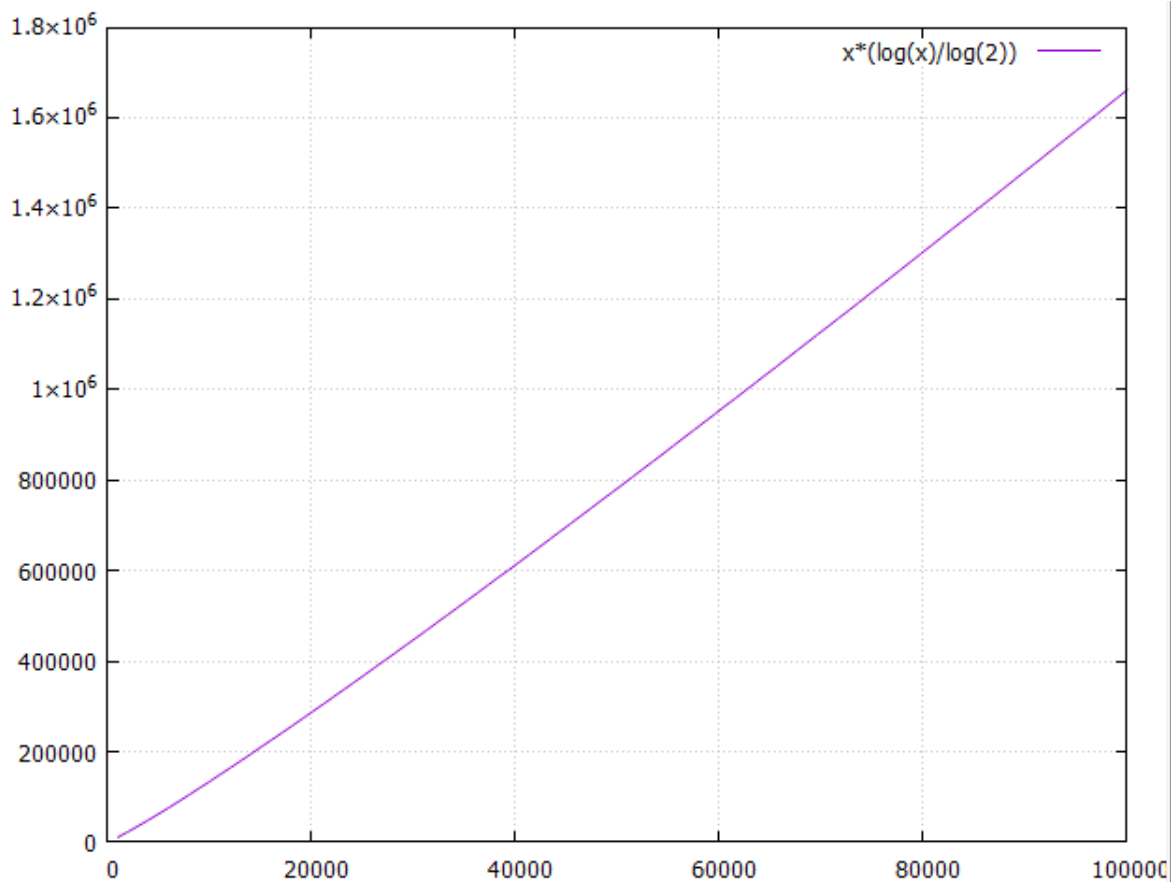
Gráficas en rango $0 < n < 100,000$

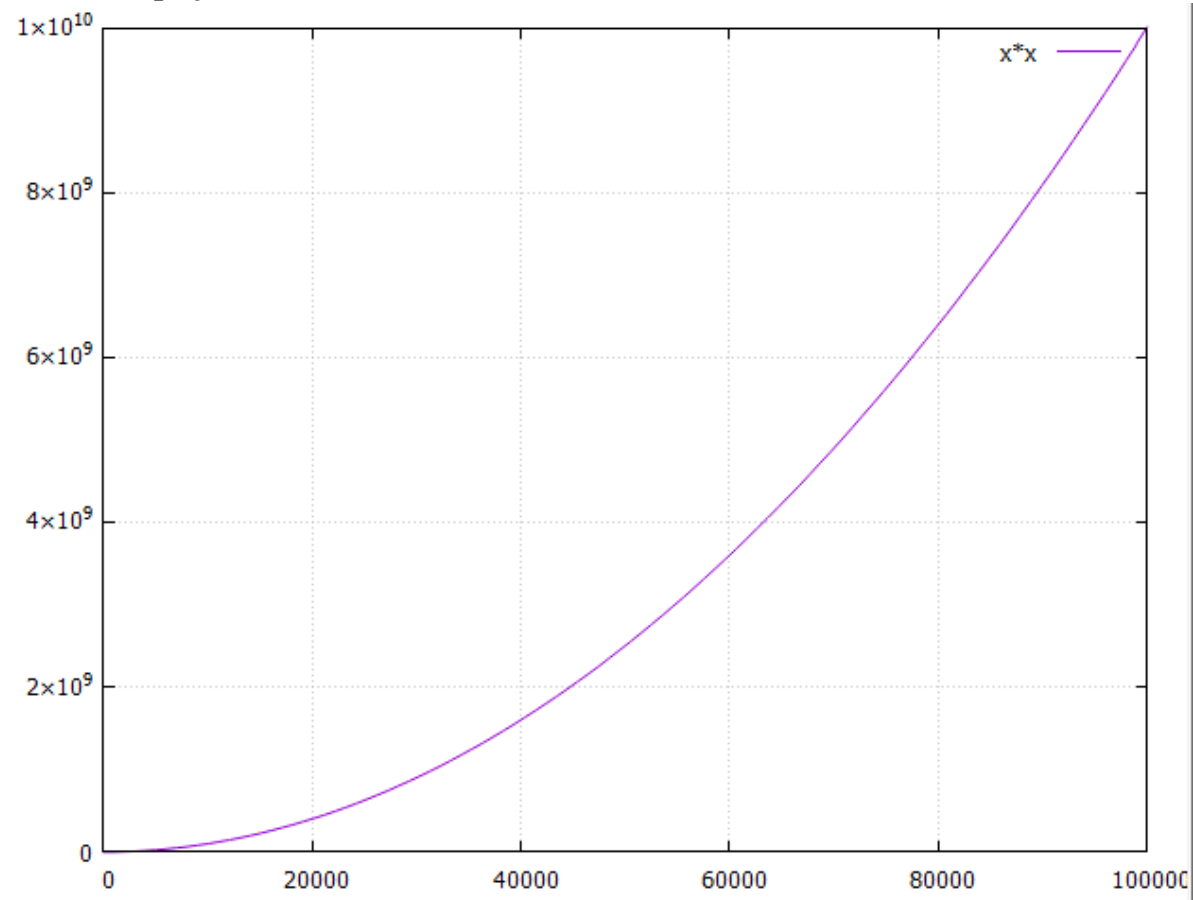
A continuación, se presentan las gráficas por separado de cada orden de complejidad dado, de acuerdo con un rango establecido.

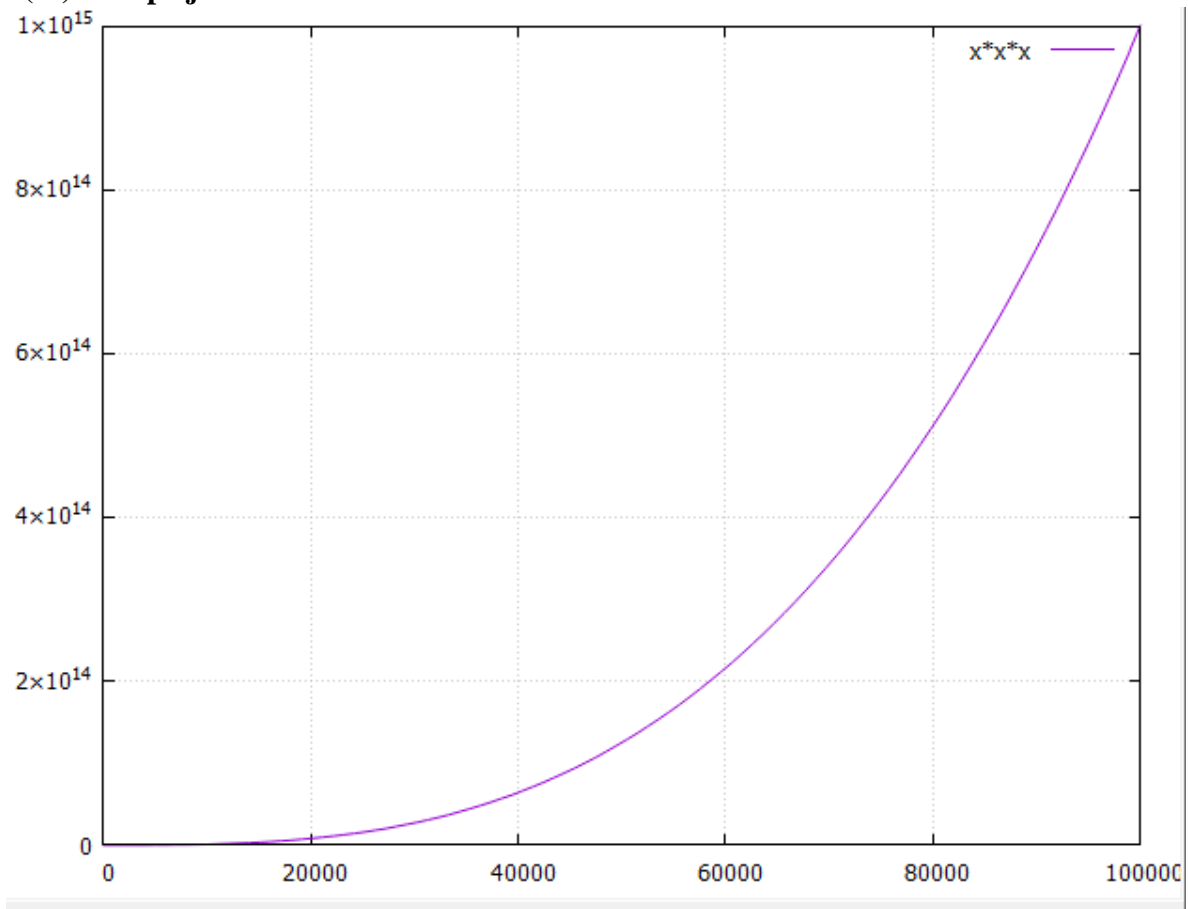
 $O(1)$ Complejidad constante

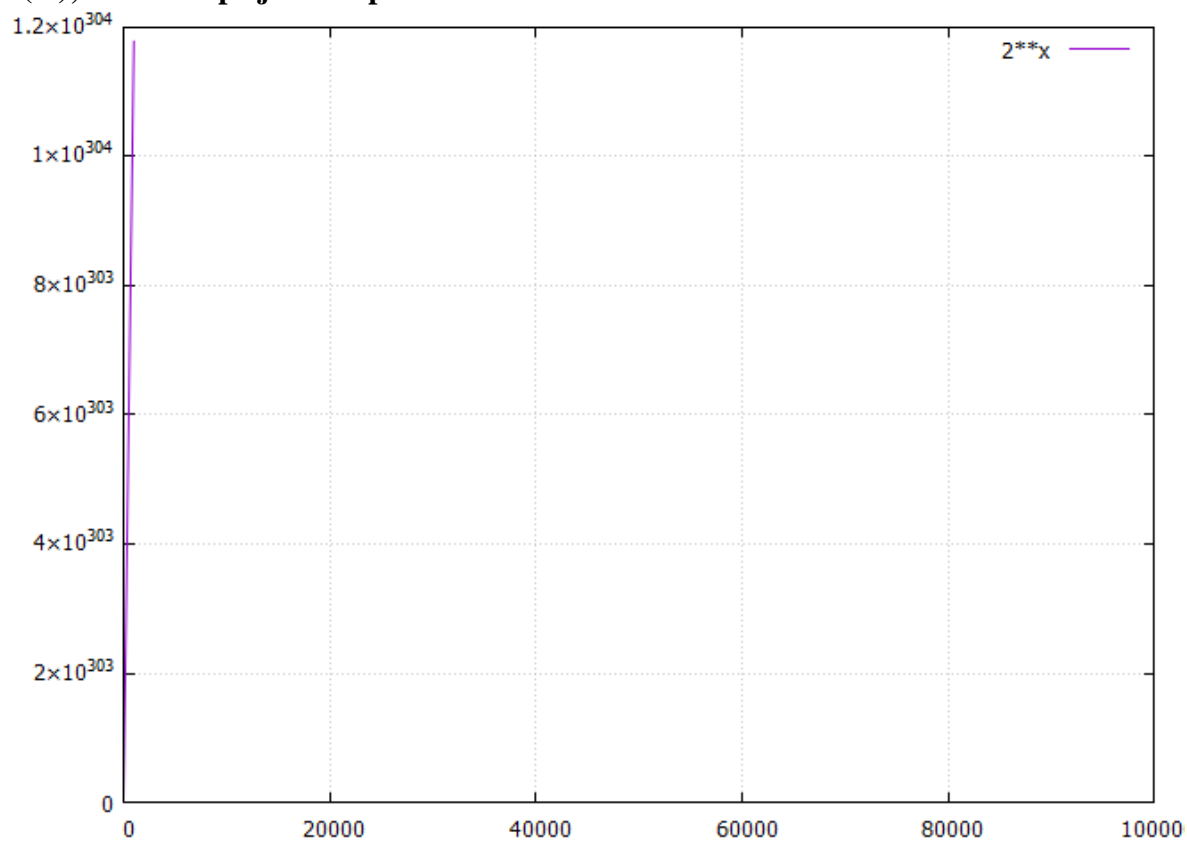
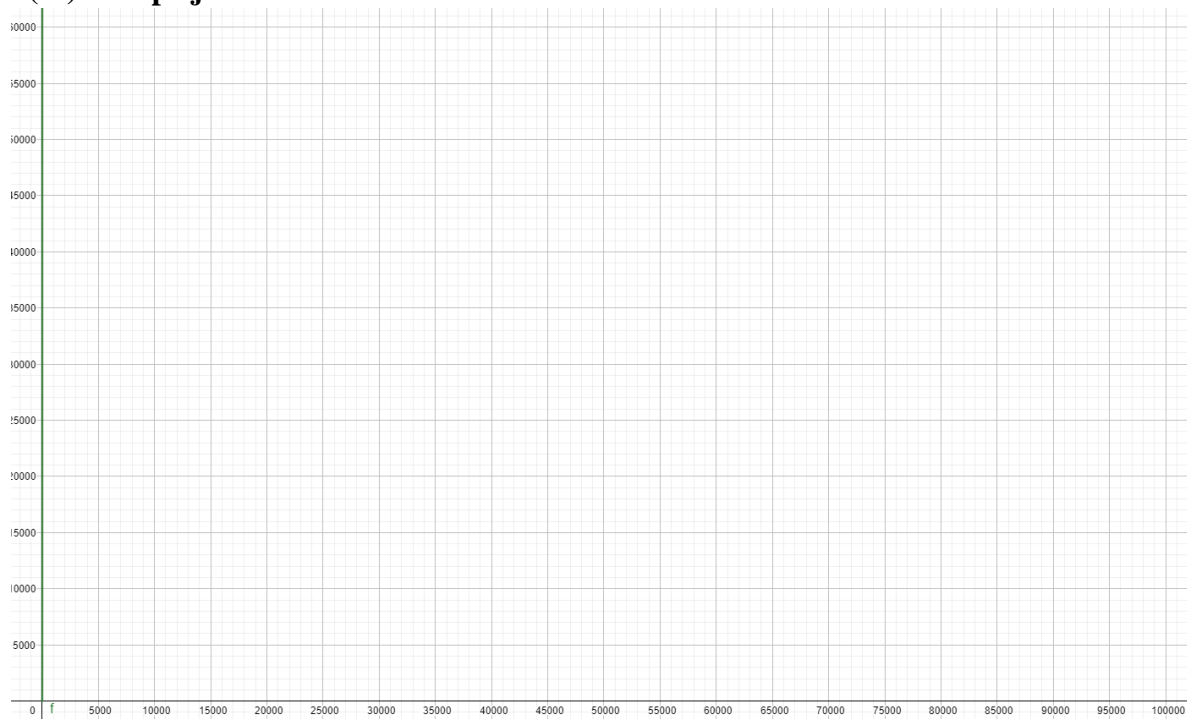
$O(\log(n))$ Complejidad logarítmica

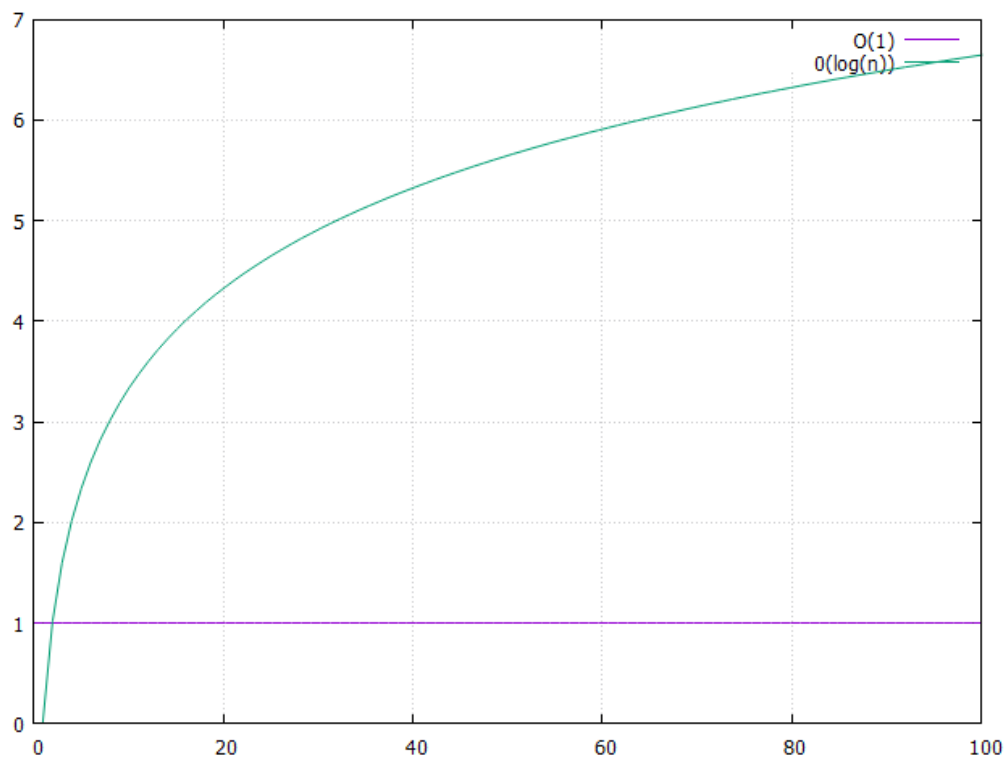
$O(n)$ Complejidad lineal

$O(n \log n)$ Complejidad “n log n”

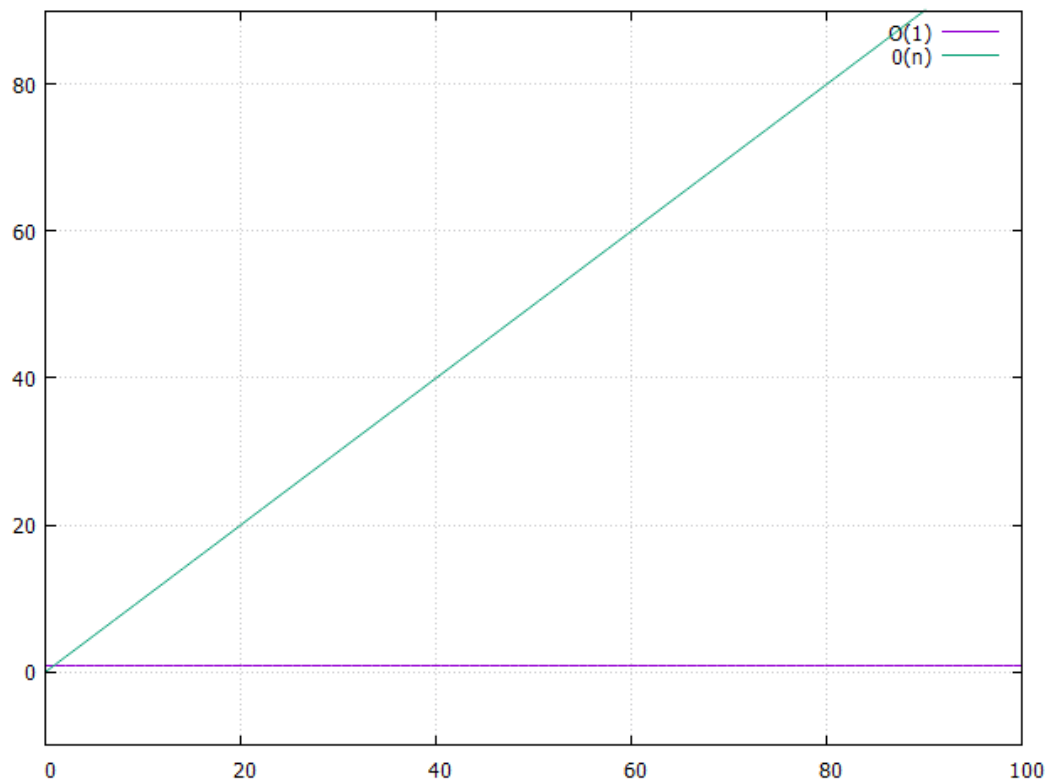
$O(n^2)$ Complejidad cuadrática

$O(n^3)$ Complejidad cubica

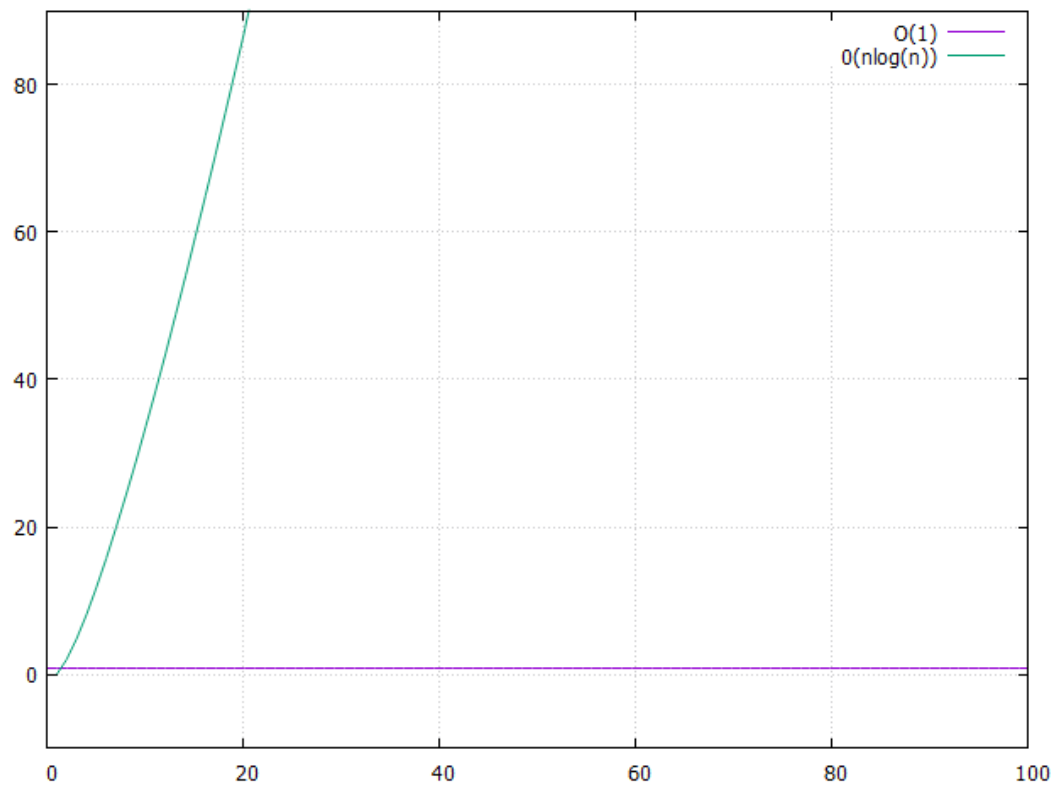
$O(c^n); c > 1$ Complejidad exponencial **$O(n!)$ Complejidad factorial**

Gráficas por pares en rango $0 < n < 100$ **$O(1)$ vs $O(\log(n))$** 

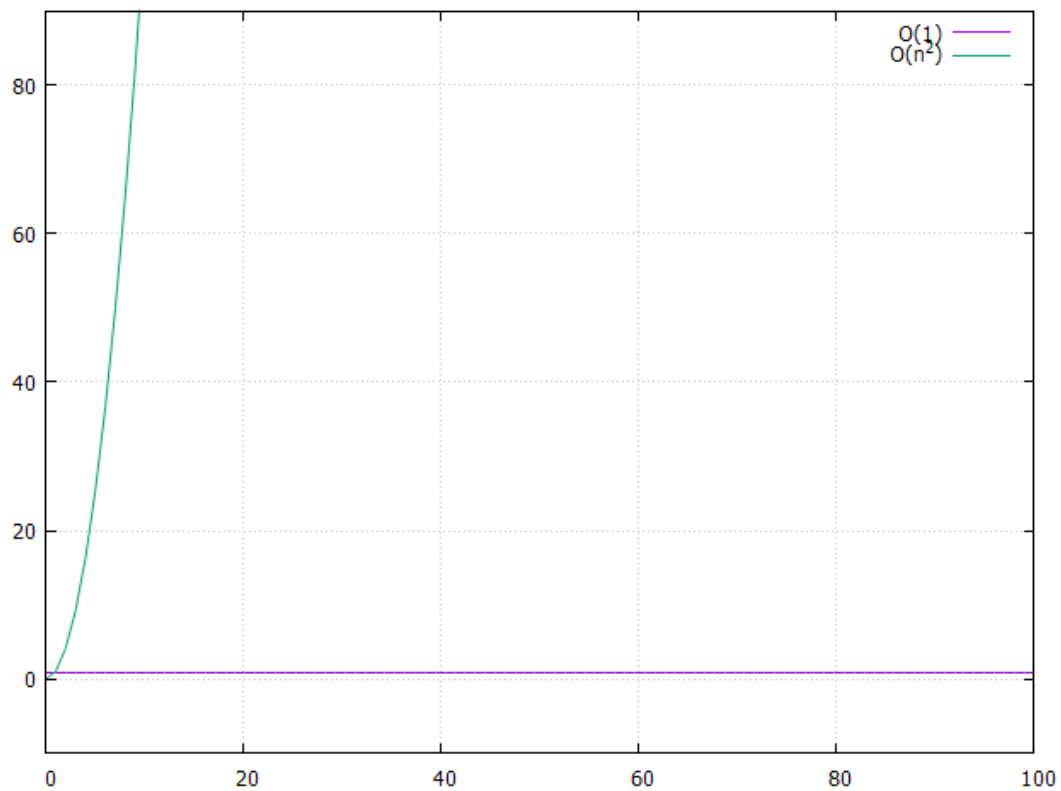
En esta parte escogería la de orden constante, ya que como su nombre lo dice, es constante y sería mejor que la de orden logarítmico.

$O(1)$ vs $O(n)$ 

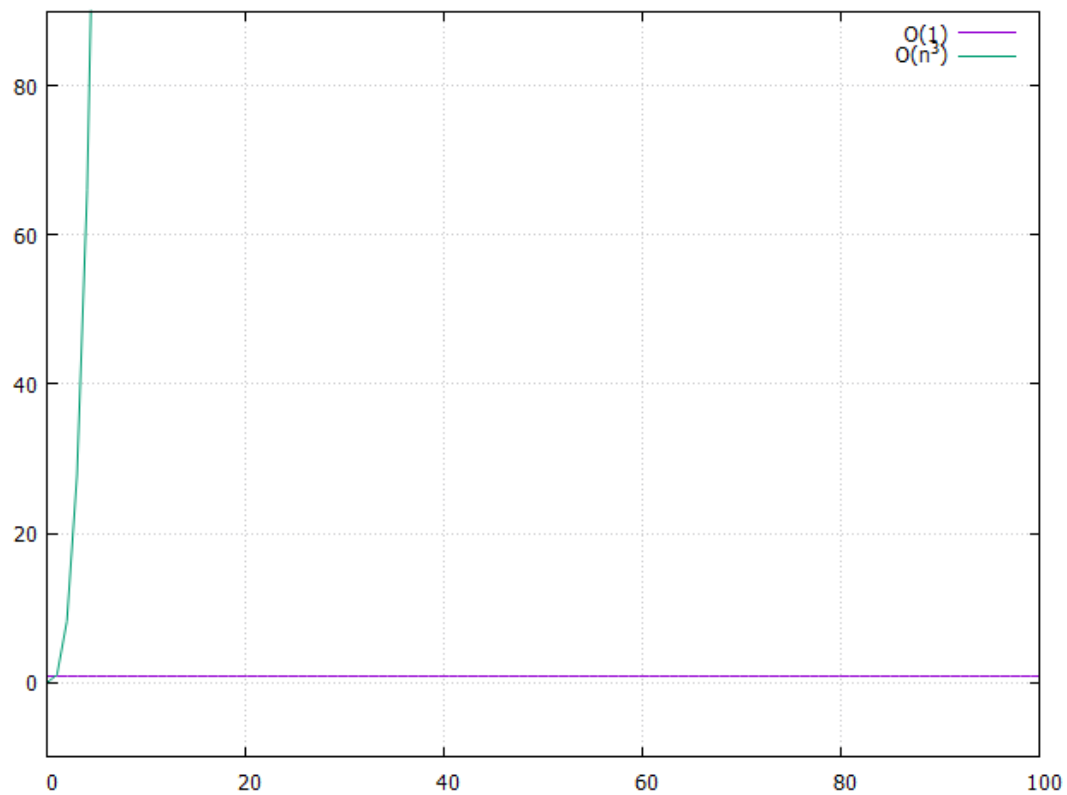
La mejor de ambas y la que tiene un mejor comportamiento es la constante

$O(1)$ vs $O(n\log(n))$ 

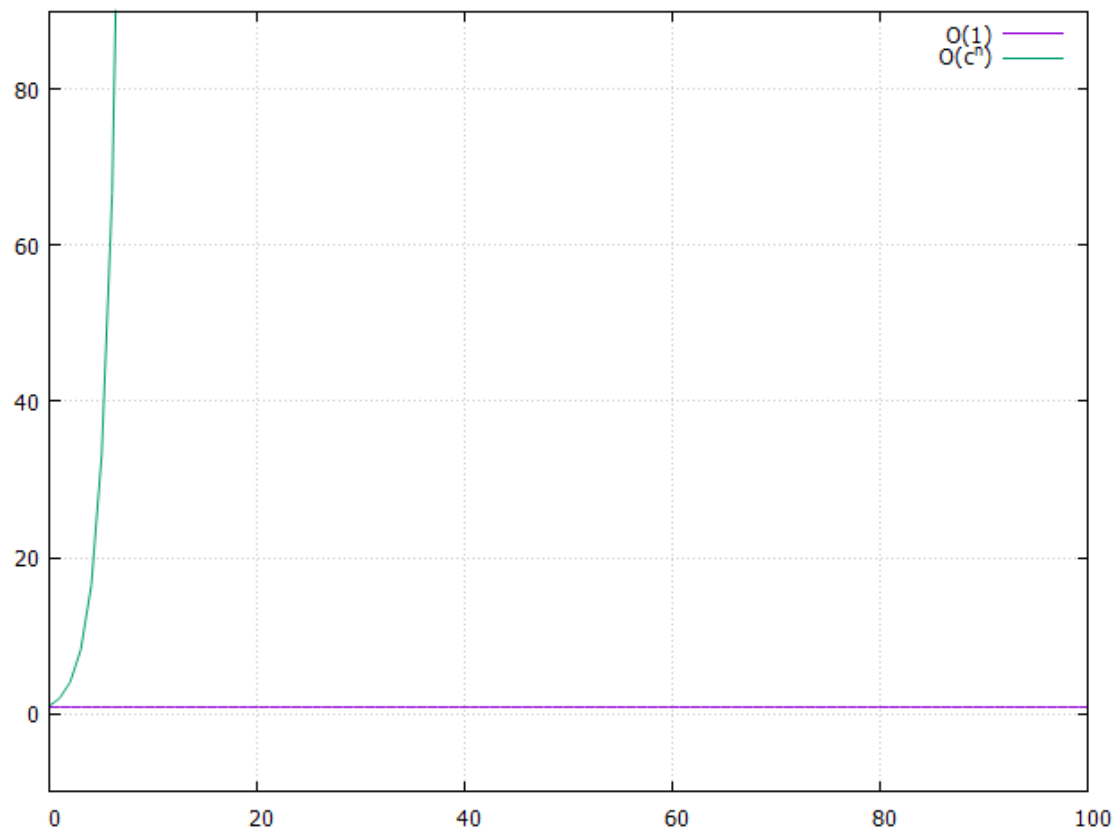
De este par, al igual que en las anteriores la mejor de las dos es la constante por lo tanto es la que elegiría.

$O(1)$ vs $O(n^2)$ 

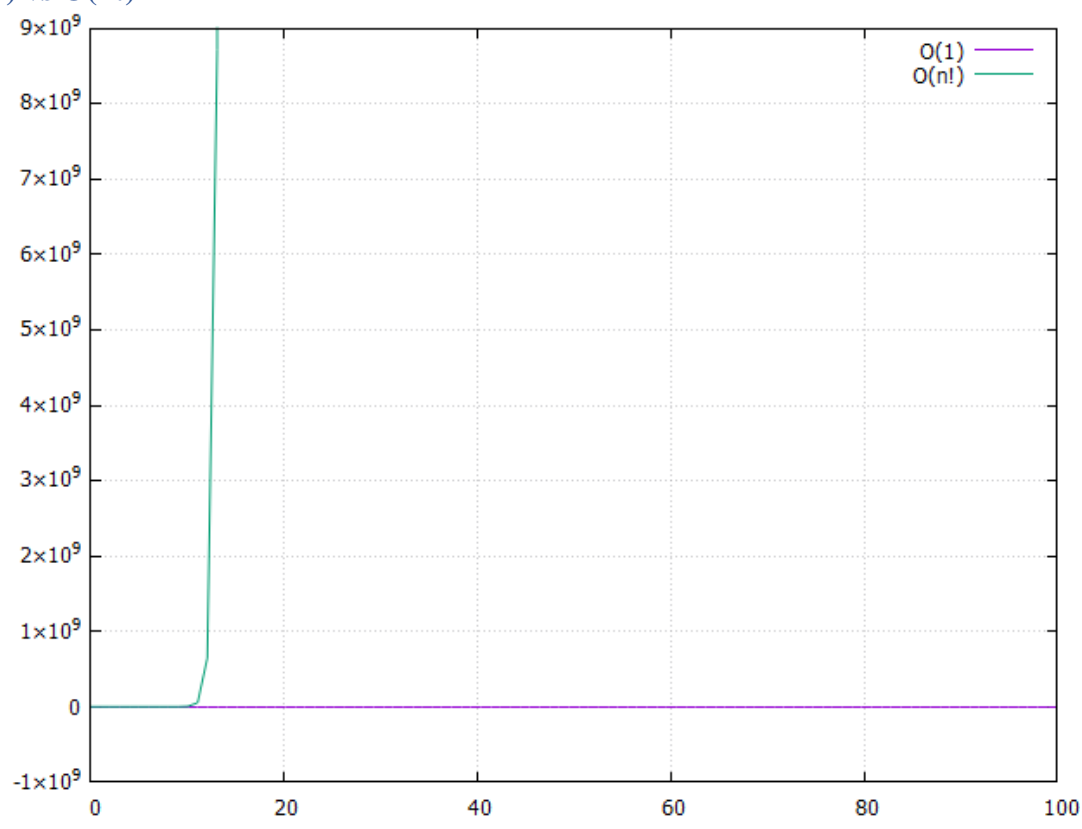
La función de orden 1 es la que elegiría sobre la función de orden cuadrático, podemos ver en la gráfica que su comportamiento es mejor que la cuadrática.

$O(1)$ vs $O(n^3)$ 

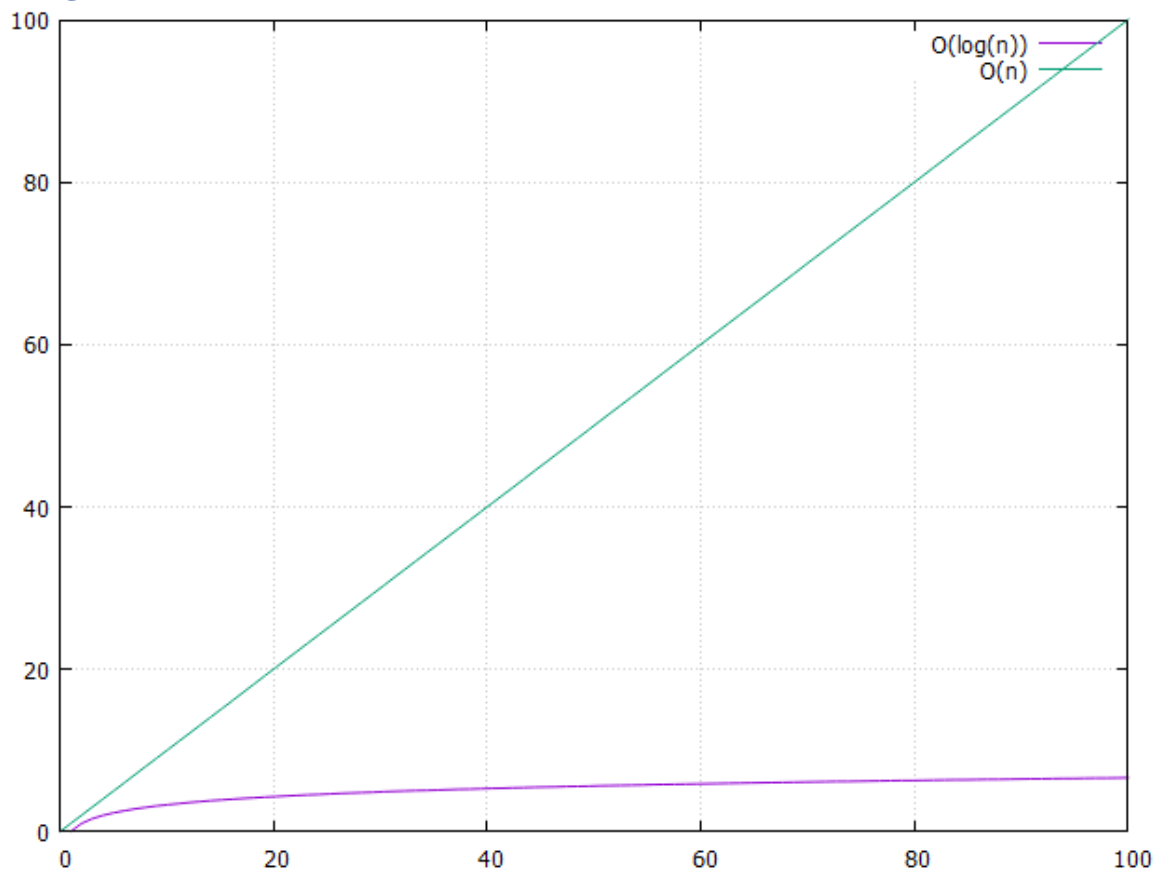
De manera similar que la anterior, se elige la de orden constante

$O(1)$ vs $O(c^n)$ 

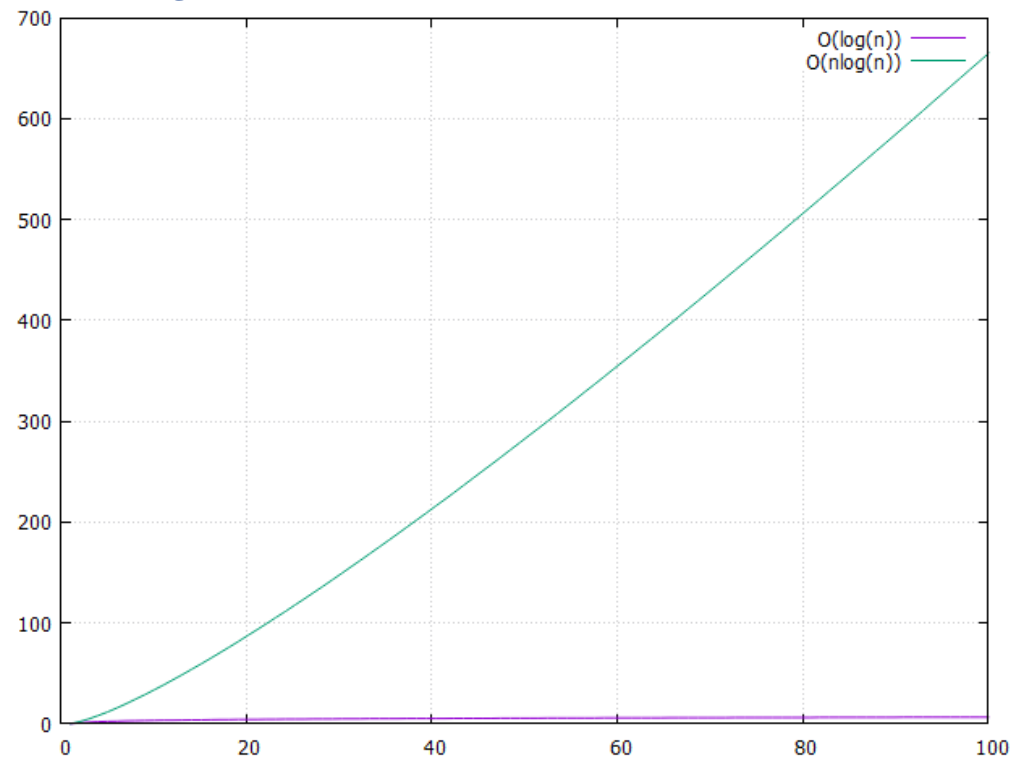
El orden c^n es de los peores casos que puede haber en un algoritmo, por lo tanto la mejor opción es la de orden constante 1.

O(1) vs O(n!)

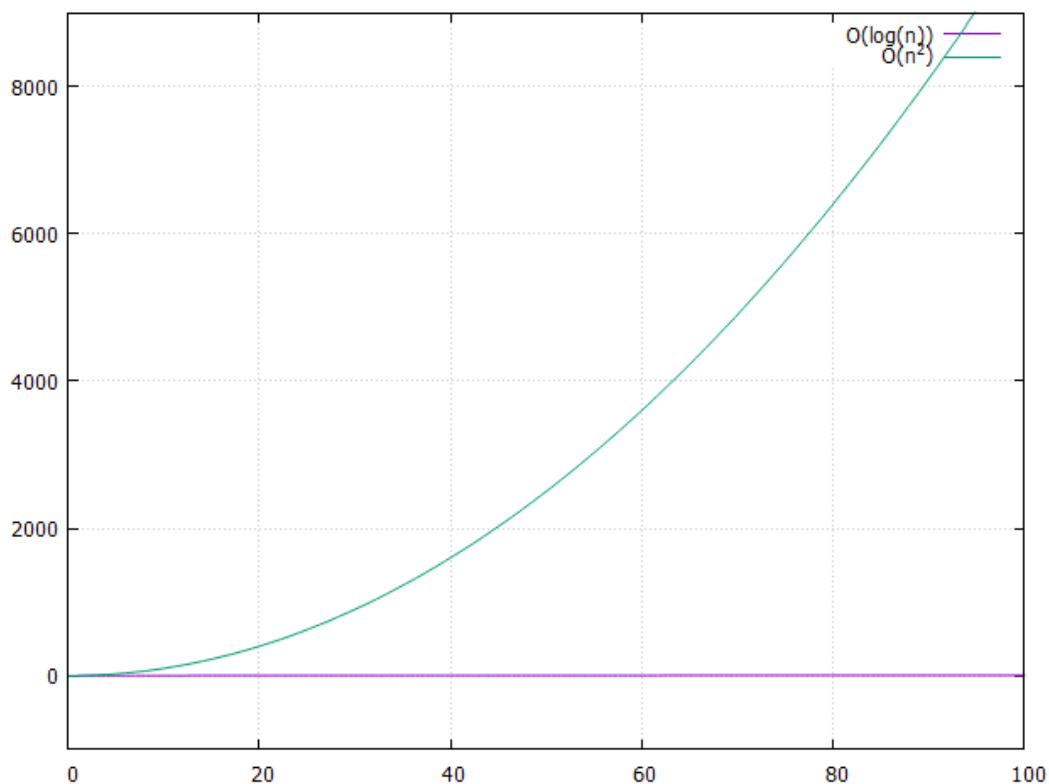
Los algoritmos con orden $n!$ son los peores que puede haber, pues como se ve en la grafica tienen un comportamiento muy caro, al igual que los casos pasados, se escoge la función de orden 1.

$O(\log(n))$ vs $O(n)$ 

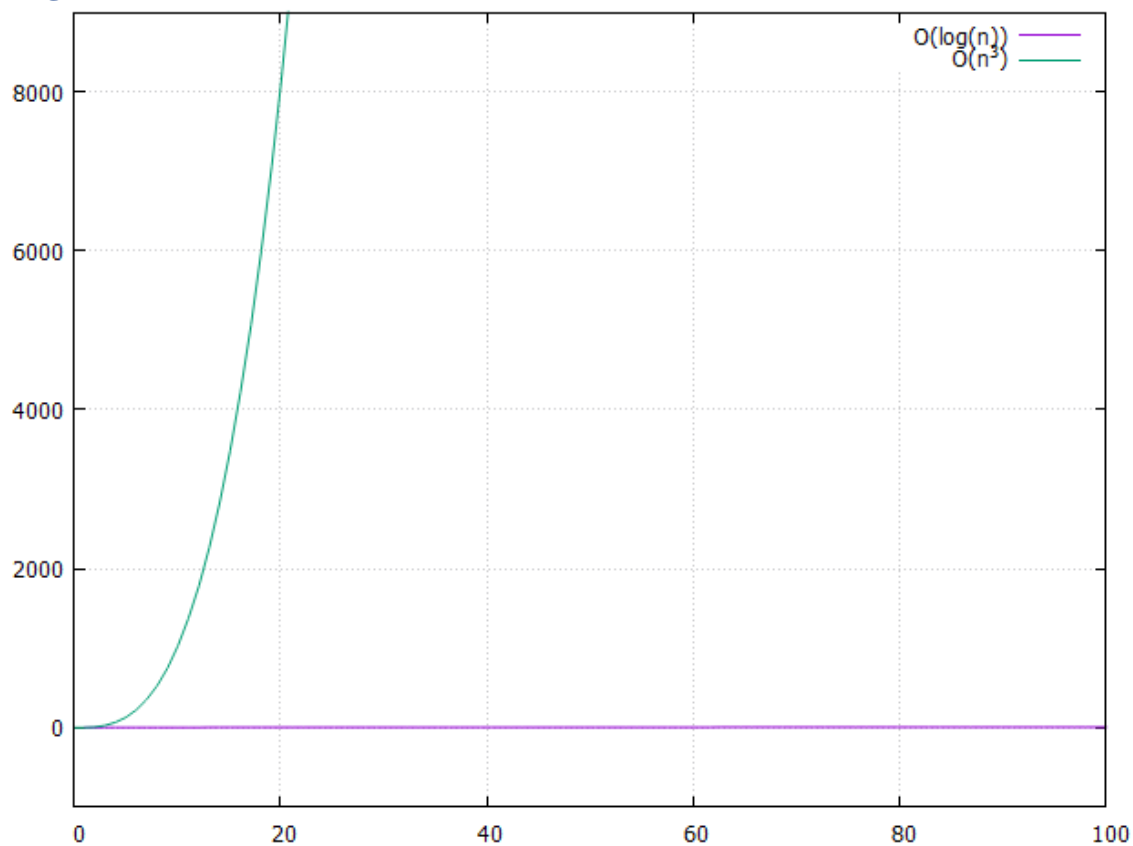
Vemos que la peor de las dos gráficas ahora es la de grado n , por lo tanto escogería $O(\log(n))$.

$O(\log(n))$ vs $O(n\log(n))$ 

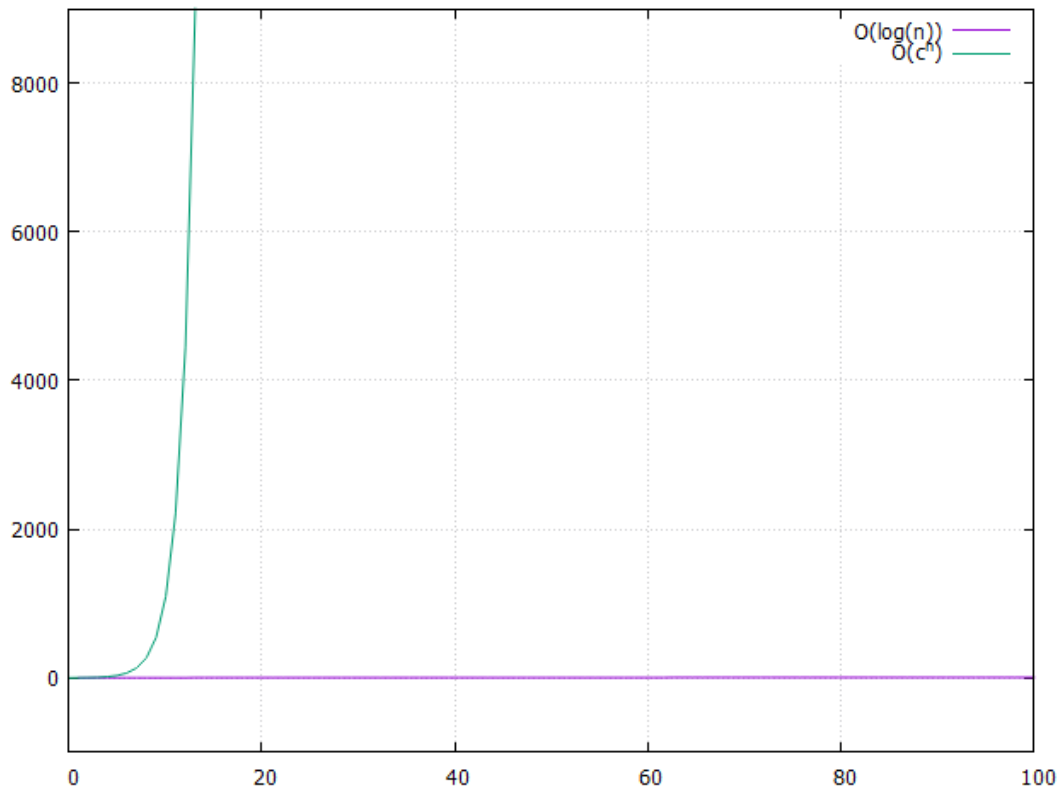
Al igual como pasaba con la función de orden constante 1, en esta comparación se ve como el orden logarítmico está muy por debajo por la de orden $n\log n$, siendo la de orden $\log n$ mejor.

$O(\log(n))$ vs $O(n^2)$ 

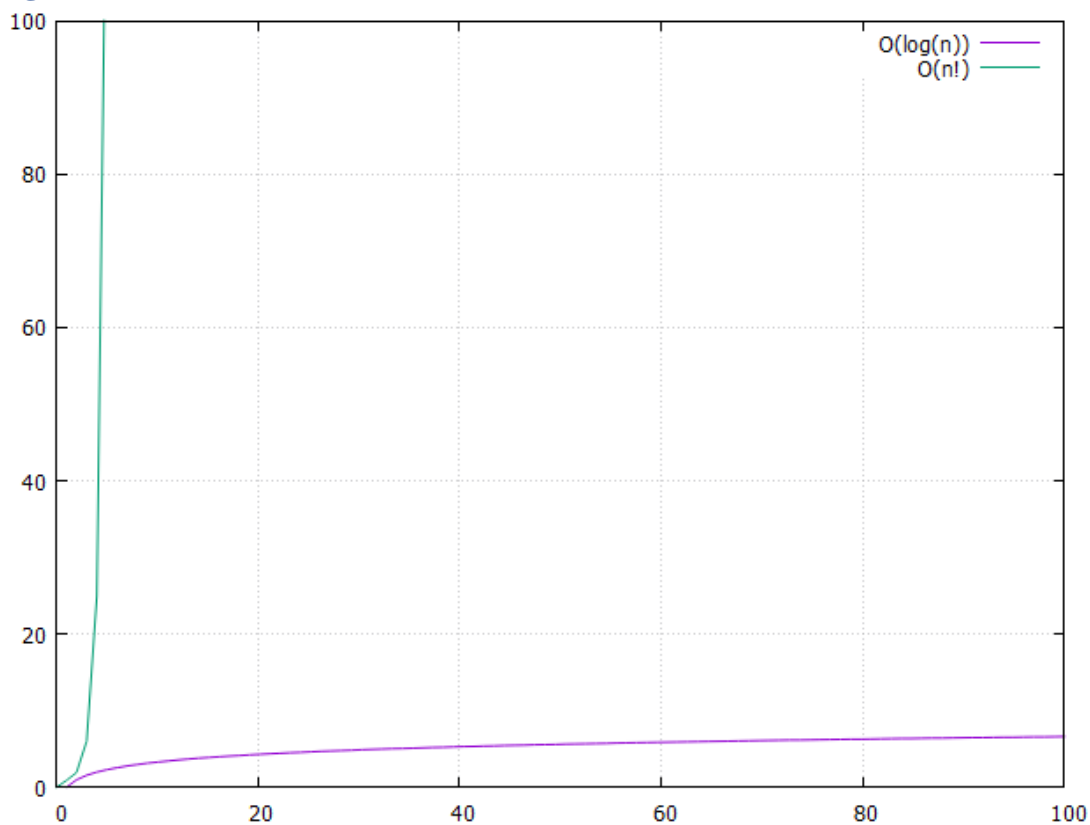
Vemos como al principio van teniendo un comportamiento similar, pero no tarda la de grado cuadrado en crecer mucho más rápido que la de orden $\log n$, por esa razón elegiría la de $O(\log(n))$.

$O(\log(n))$ vs $O(n^3)$ 

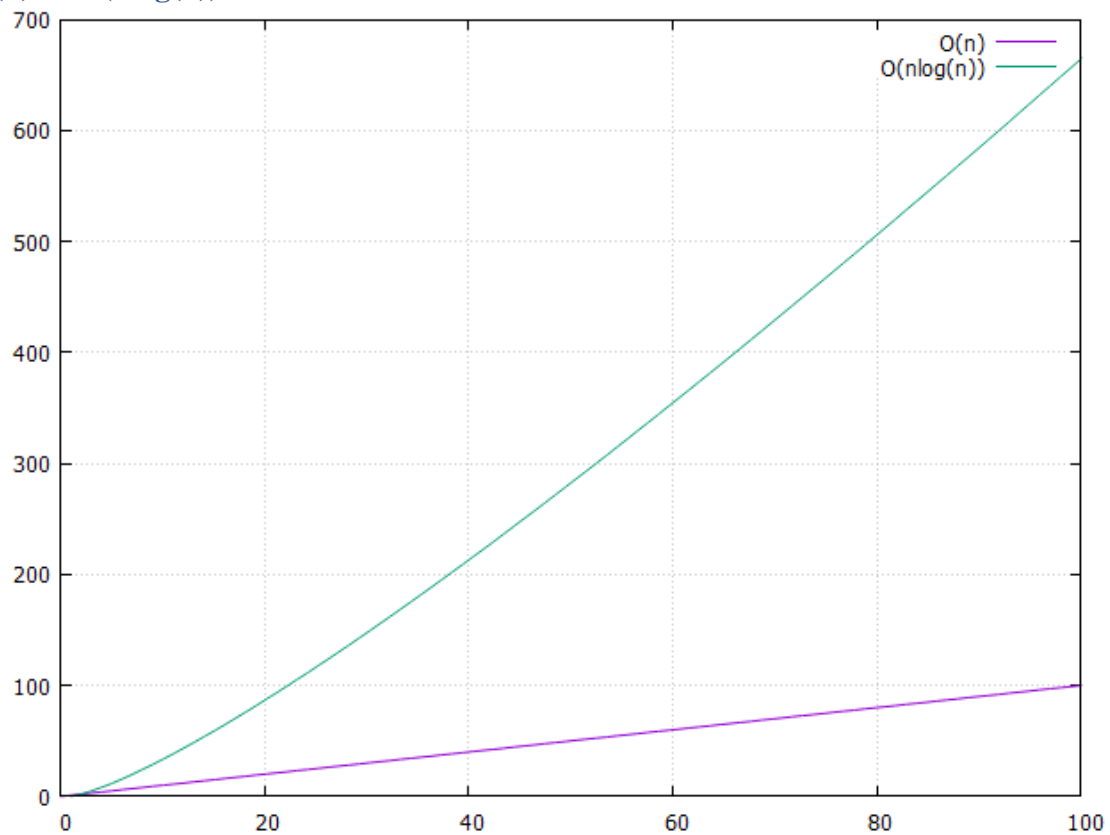
Podemos observar como la funcion de orden cubico crece de una manera muy rapida a comparacion de la de orden logn, siendo mucho mejor la función de orden logaritmico.

$O(\log(n))$ vs $O(c^n)$ 

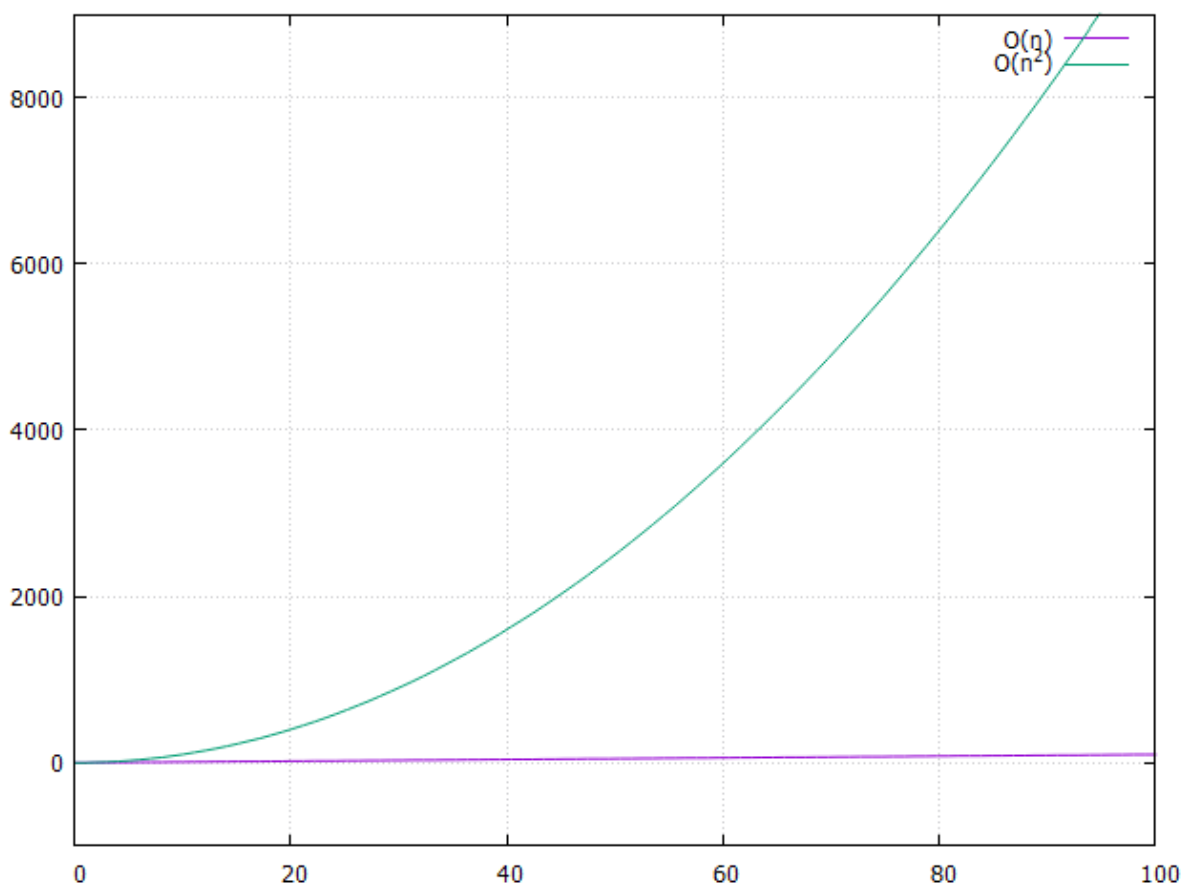
Aquí como sucede en el caso anterior, la función de orden c^n tiene un comportamiento exponencial, por lo tanto un crecimiento mucho más rápido. en contraste vemos la de orden logaritmico la cual a este punto parece que se comporta como una función de orden constante 1, no es así pero da la impresión, por lo tanto elegiría esta ultima de orden $\log n$.

$O(\log(n))$ vs $O(n!)$ 

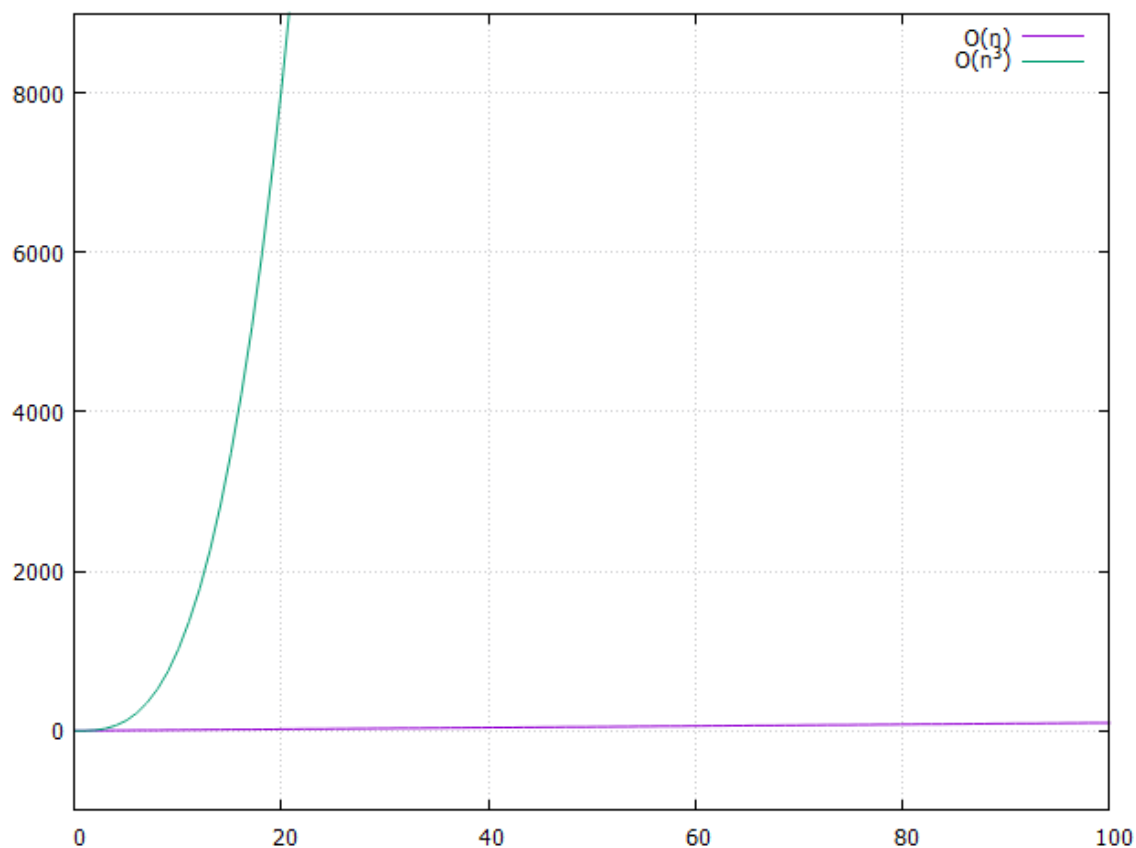
Como mencione. lo peor que te puedes encontrar es un algoritmo con orden $n!$. en este caso lo tenemos en la gráfica, por obvias razones elegiría la de orden $\log n$.

$O(n)$ vs $O(n\log(n))$ 

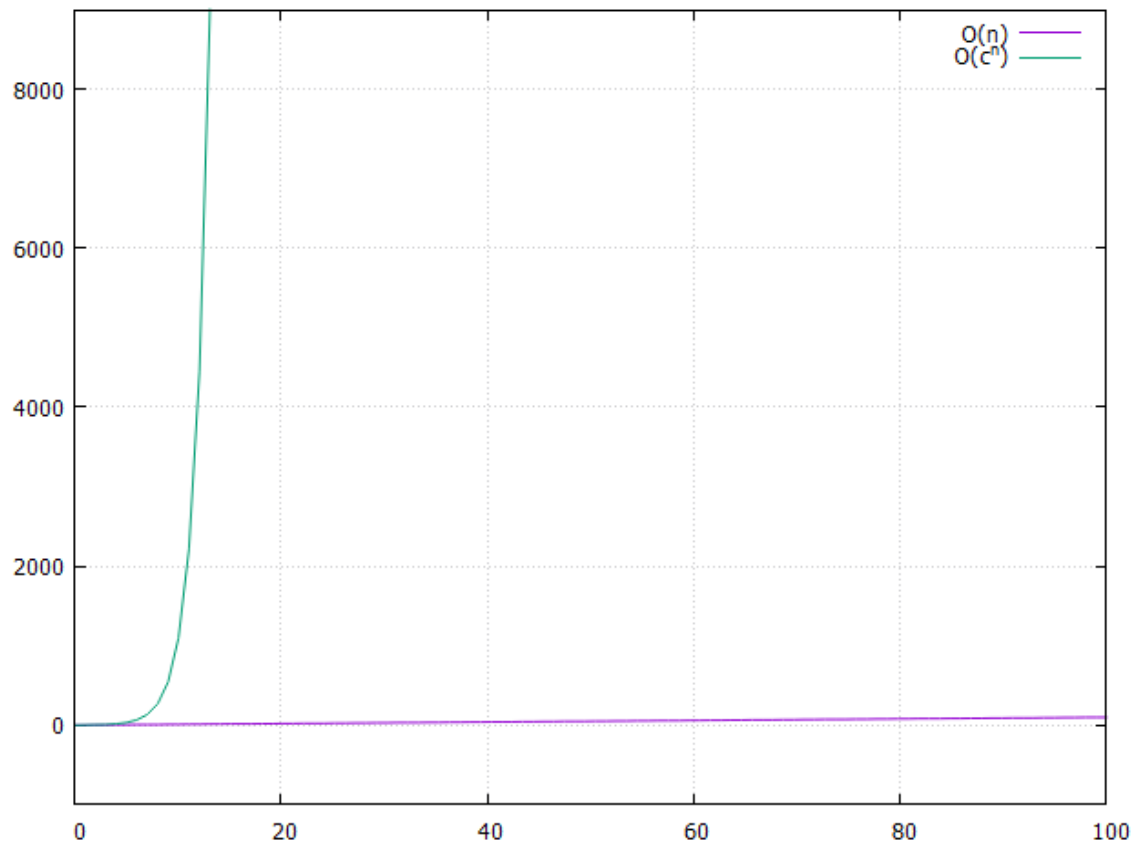
Ahora estamos viendo como la función de orden n es mucho mejor comparada con la de orden $n\log n$, por lo tanto elegiría la de orden n .

$O(n)$ vs $O(n^2)$ 

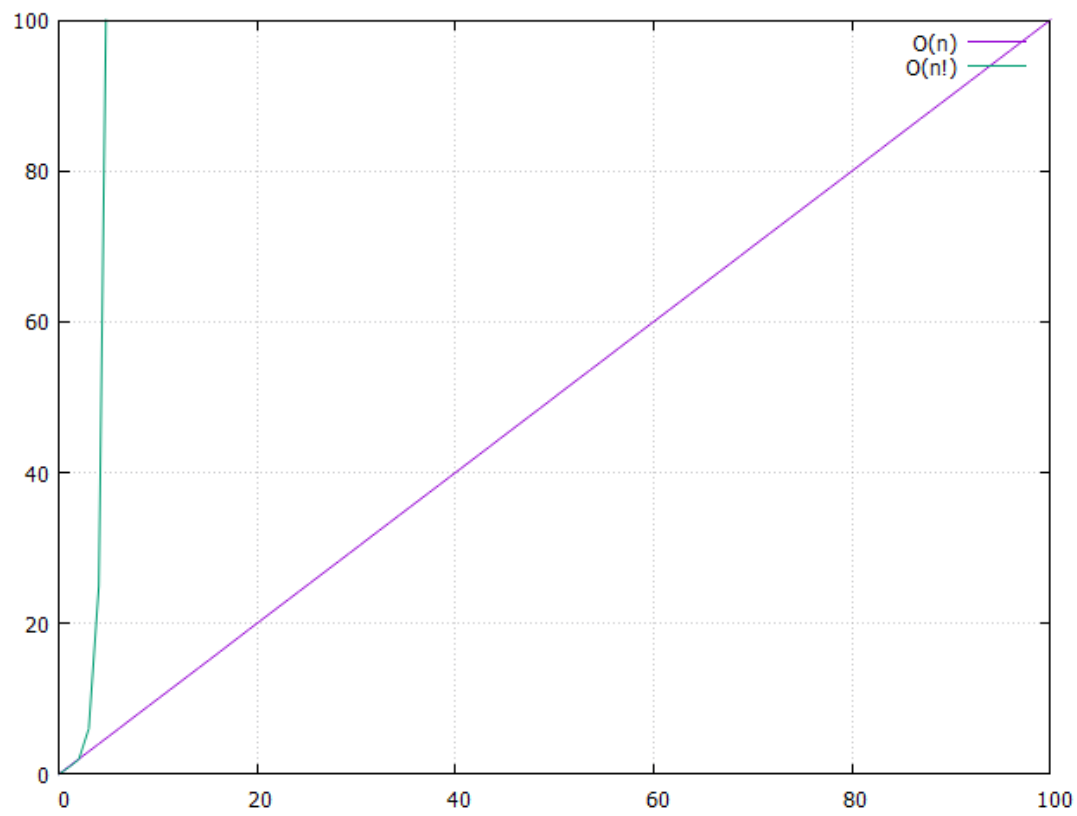
En esta comparación escogería la de orden n sobre la de orden n^2 pues la diferencia se ve notoria en la gráfica al punto en que la de orden n parece tener un comportamiento constante.

$O(n)$ vs $O(n^3)$ 

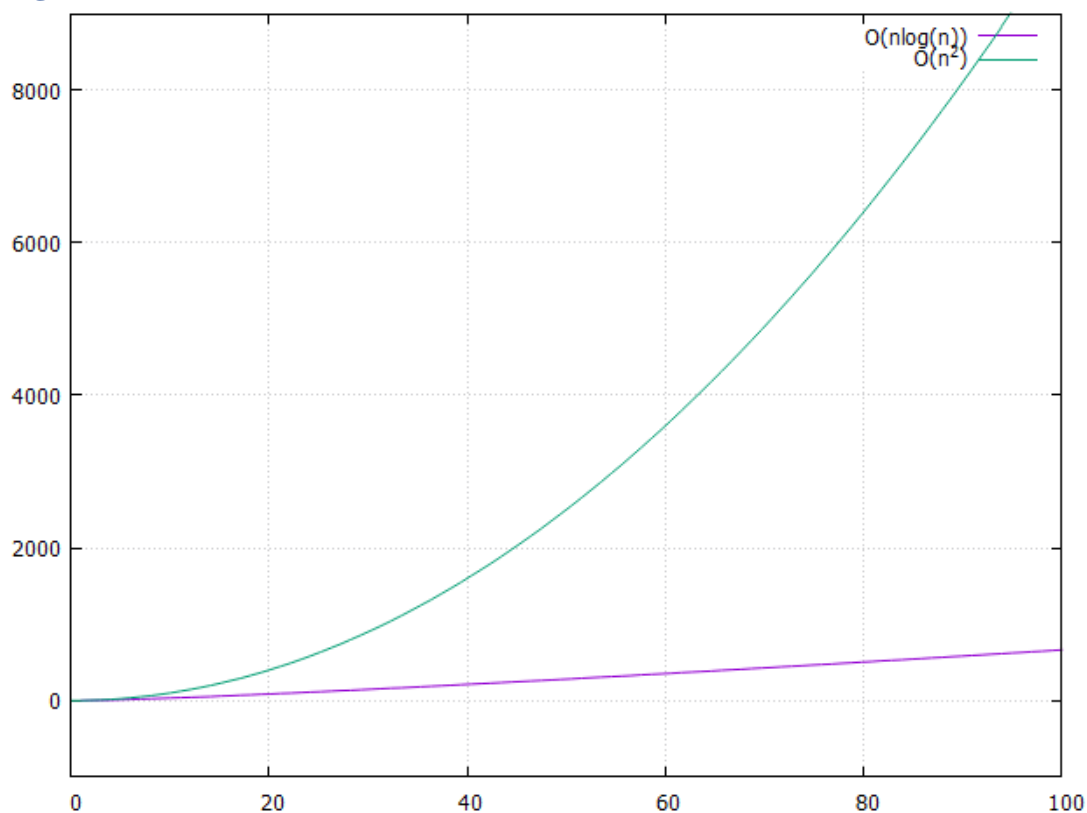
Como pasa en el anterior ejemplos, en esta comparación la de orden n parece tener un comportamiento lineal respecto a la de orden n^3 , por lo tanto escogería la de orden n .

$O(n)$ vs $O(c^n)$ 

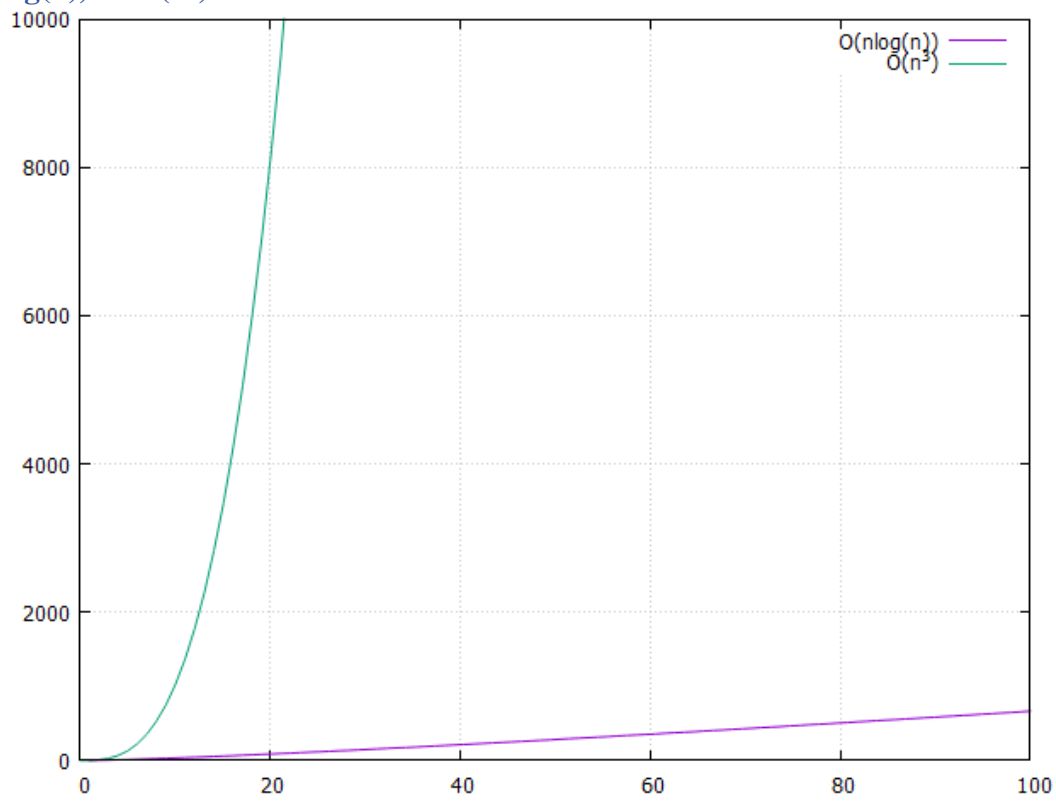
Aquí por obvias razones la mejor y la que escogería es la de orden n .

$O(n)$ vs $O(n!)$ 

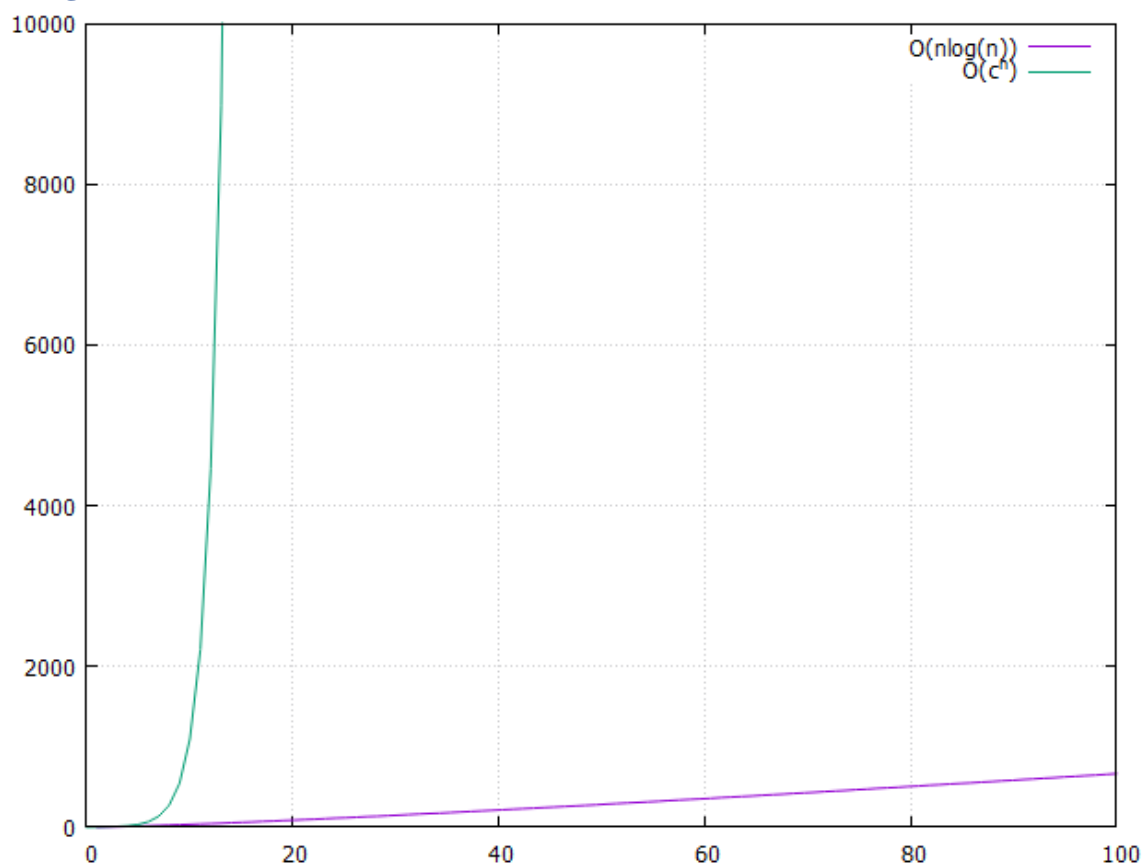
Aquí se ve claramente como la grafica de $n!$ crece de una amnera muy rapida a comparacion de la de orden n , de esta forma es facil saber cual elegir, la de orden n .

$O(n \log(n))$ vs $O(n^2)$ 

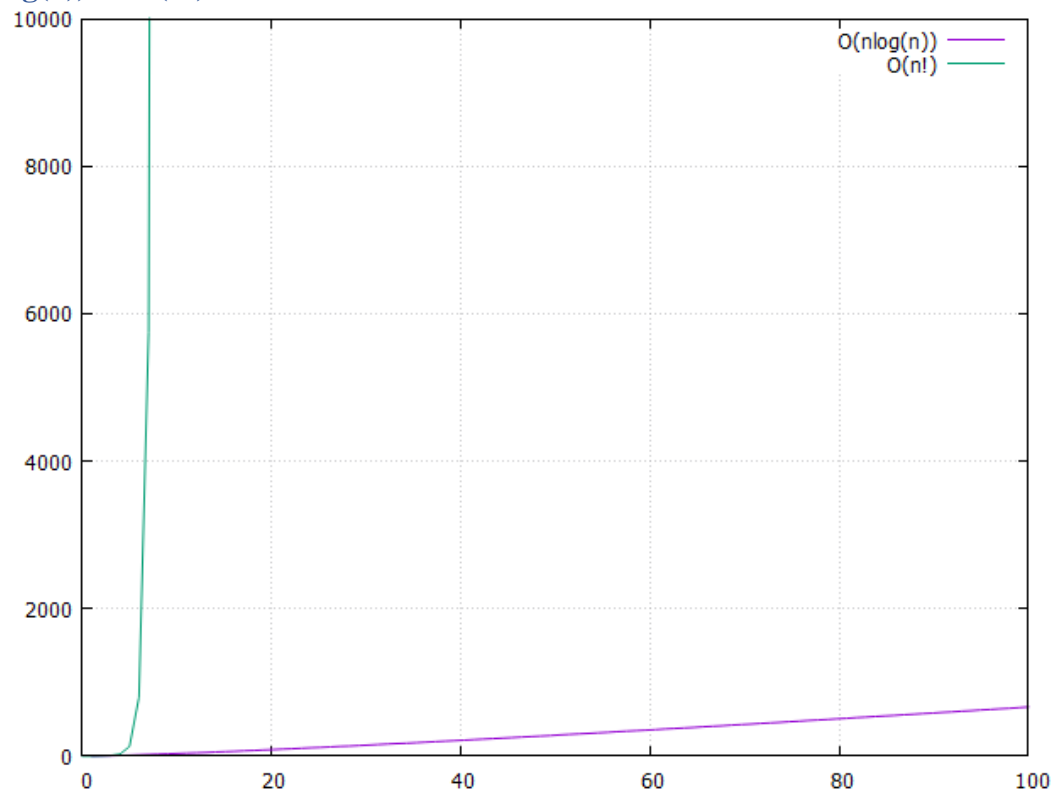
Ahora, aquí comparamos $n \log n$ respecto a n cuadrada, vemos igual la diferencia notoria entre estas en la gráfica, teniendo n cuadrada crecimiento mayor que la de $n \log n$, por lo tanto elegiría la de $n \log n$.

$O(n \log(n))$ vs $O(n^3)$ 

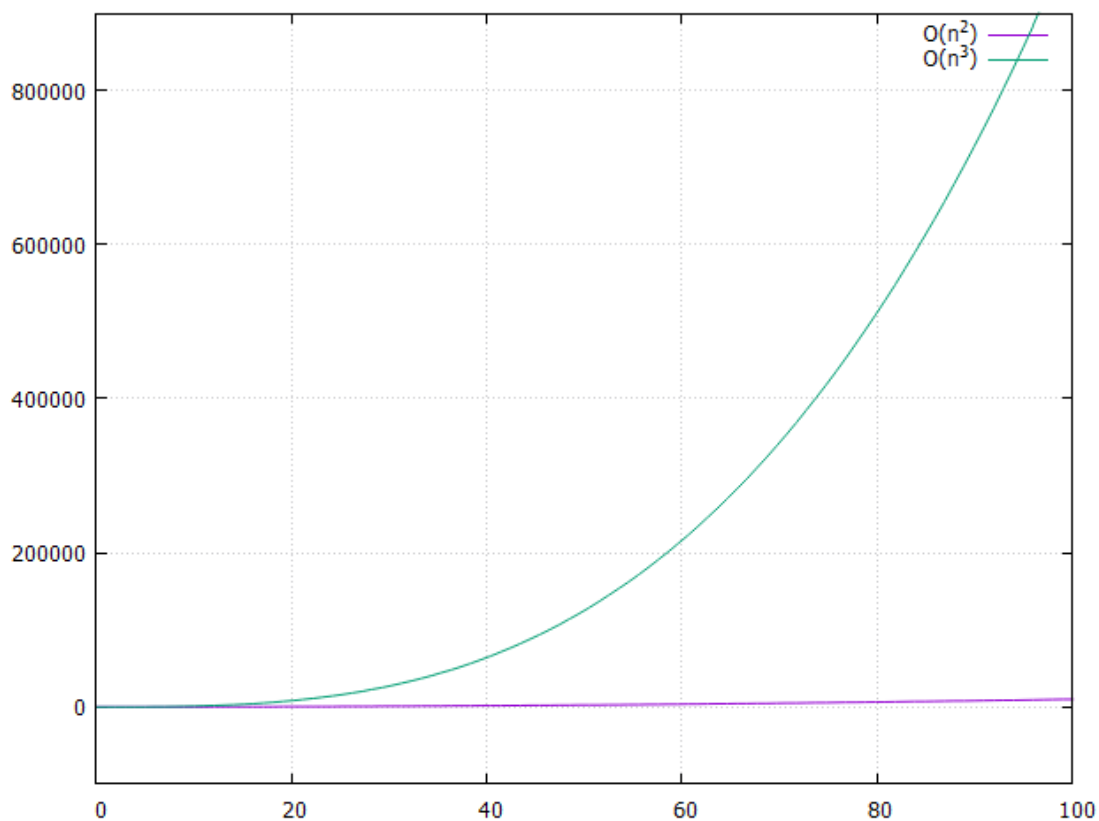
Como pasa en el anterior ejemplo, la funcion de orden n cubica es peor que la de orden nlogn, por lo tanto elegiria esta ultima.

$O(n \log(n))$ vs $O(c^n)$ 

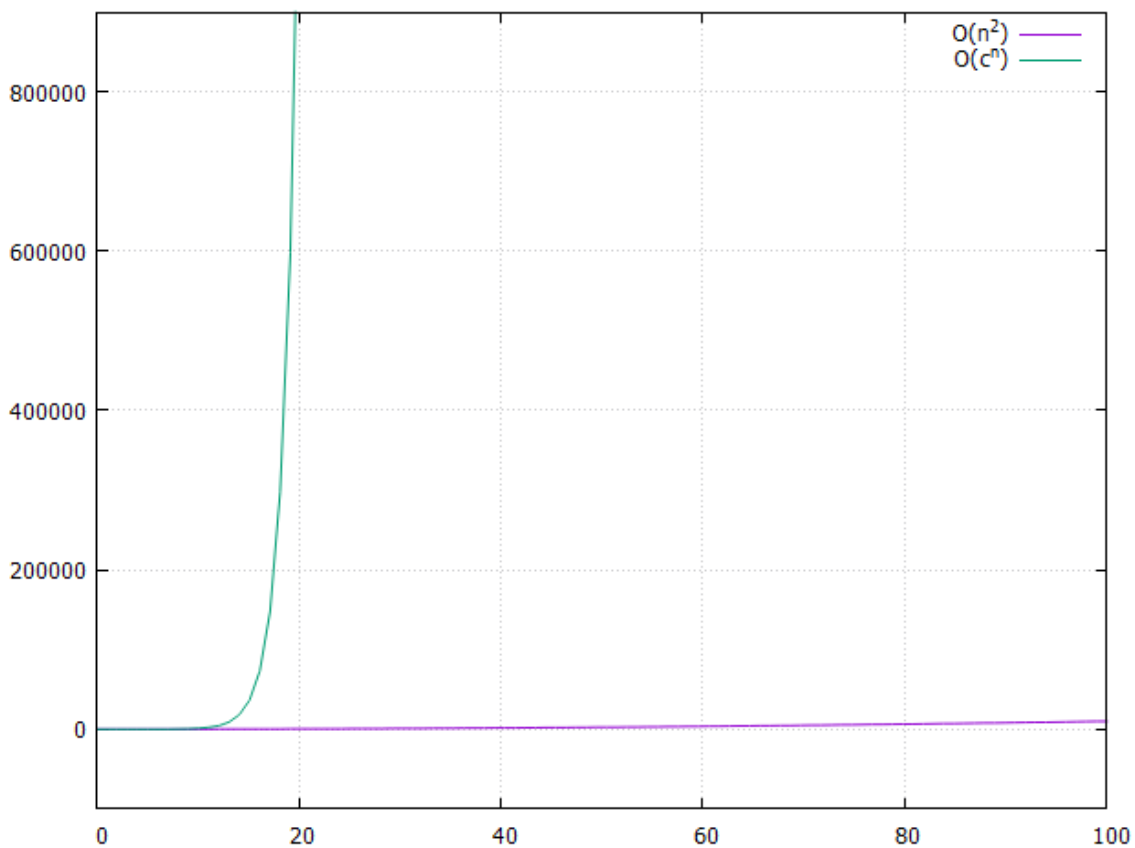
Podemos ver que el comportamiento del orden c^n es de los peores casos que se puede dar, vemos en la gráfica que la función de orden $n \log n$ esta muy por debajo, por lo tanto escogería esta.

$O(n \log(n))$ vs $O(n!)$ 

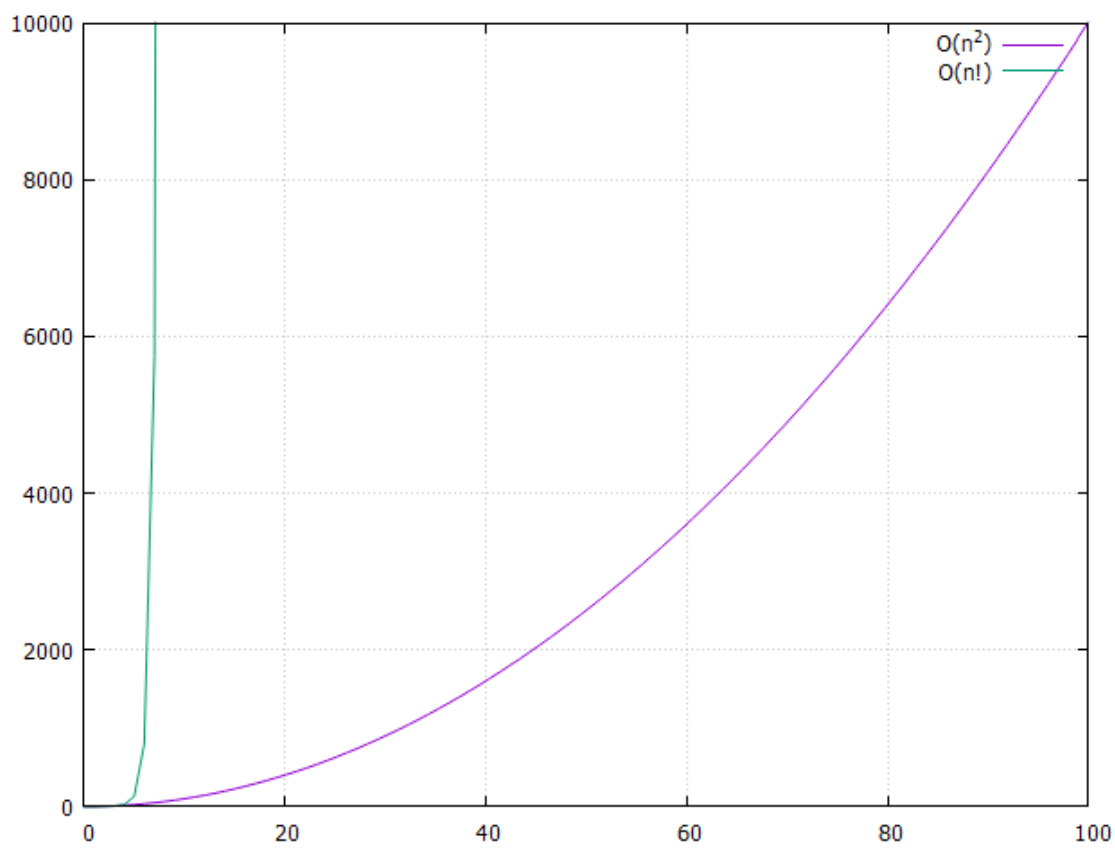
Al estar comparando contra la función de orden $n!$ que es el peor orden que se puede presentar, claramente se elige la función de orden $n \log n$.

$O(n^2)$ vs $O(n^3)$ 

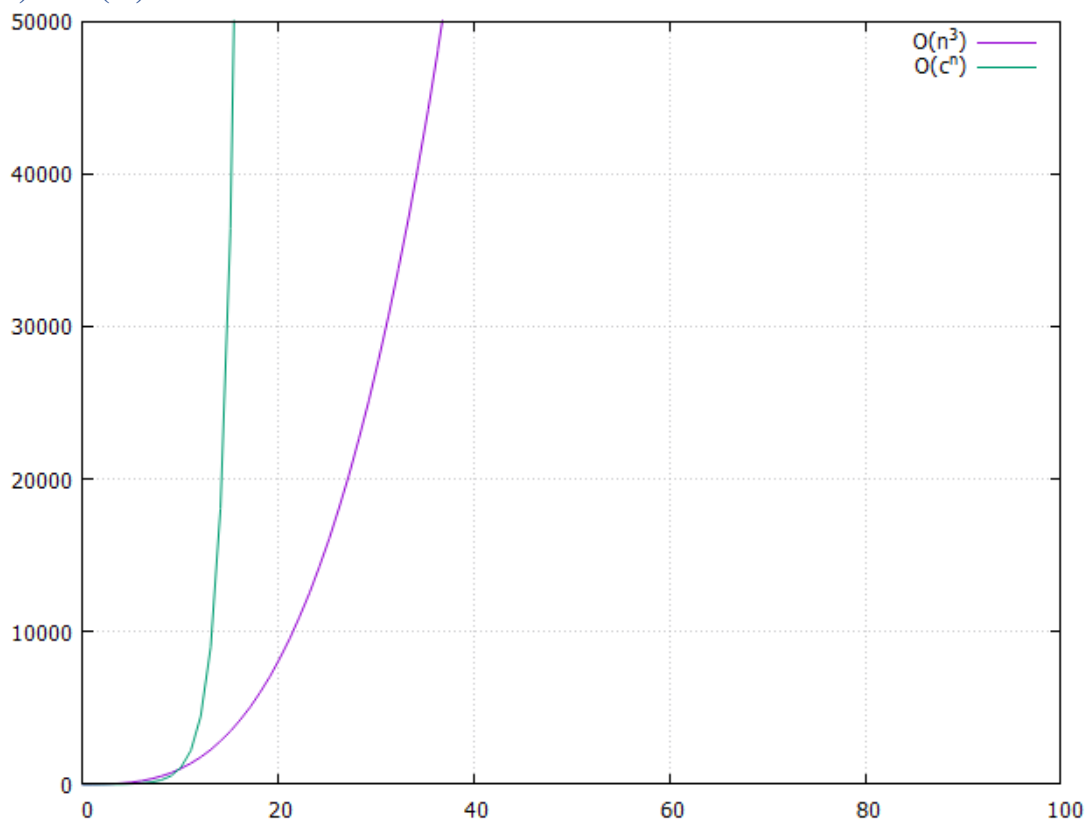
Podemos pensar que la diferencia entre n^2 y n^3 no sería mucha, pero al ver la grafica vemos que sí, la función de orden cuadrático parece comportarse de manera constante 1 a comparación de la de orden cubico, por esta razón se elige la de orden cuadrático.

$O(n^2)$ vs $O(c^n)$ 

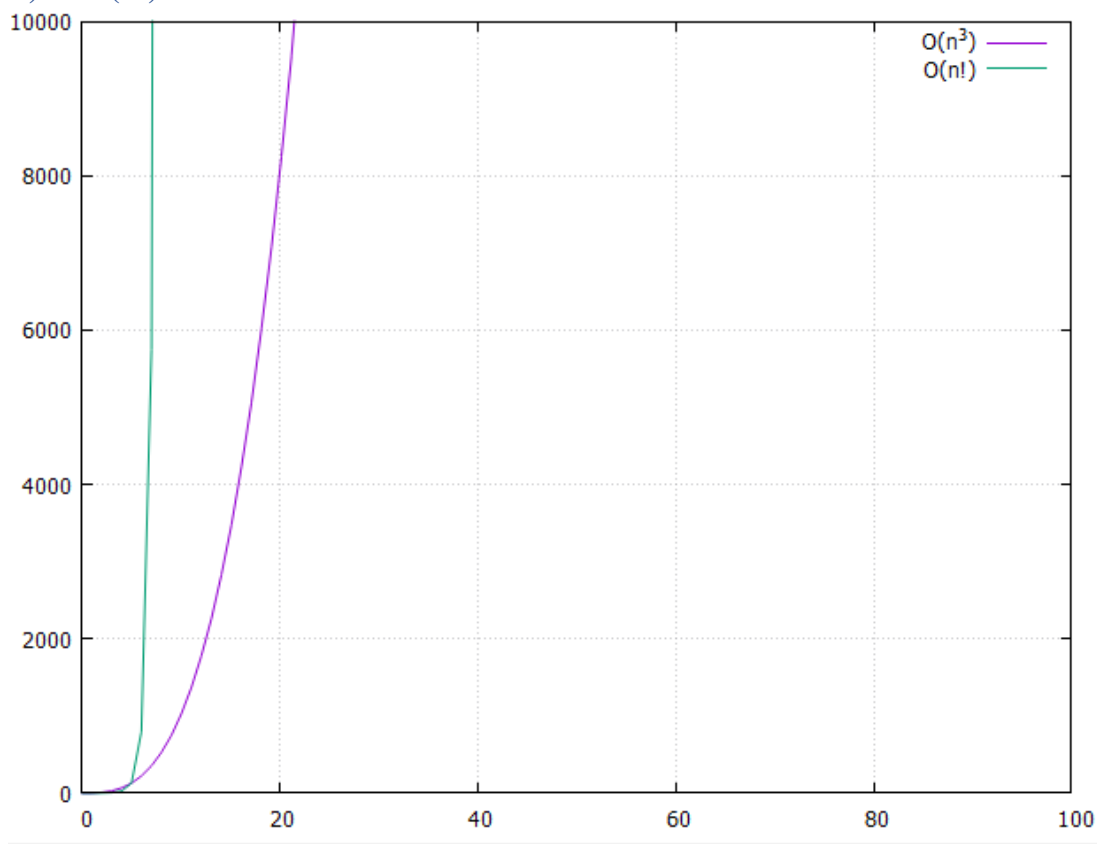
De igual forma que en el apasado ejemplo, parece que la grafica de la funcion de orden cuadrático tiene un comportamiento constante 1, sabemos que no es así, sin embargo da la impresión cuando la comparamos con la de orden c^n , eligiendo de esta forma la funcion de orden cuadrático.

$O(n^2)$ vs $O(n!)$ 

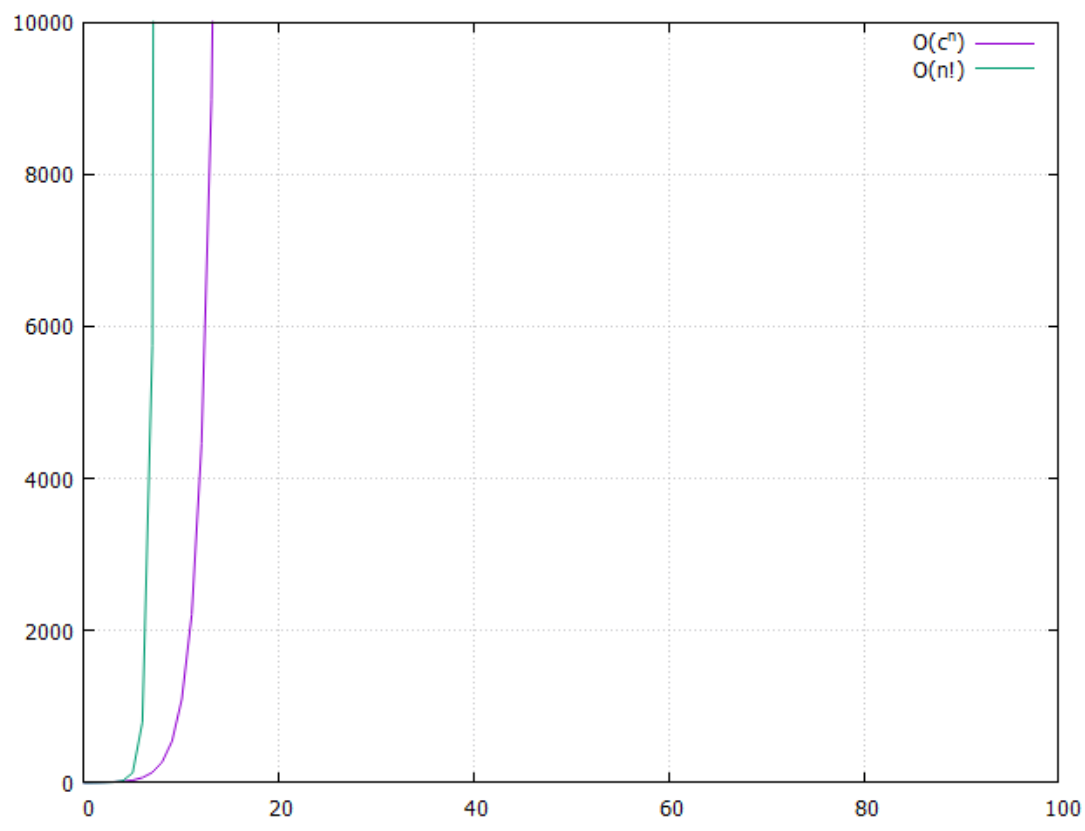
Vemos la comparación entre el orden cuadrático y factorial, como en todos los casos el de orden factorial es la que crece de una manera muy rápida, pues es el peor orden que puede haber, así, elegimos la función cuadrática.

$O(n^3)$ vs $O(c^n)$ 

Podemos ver que la función de orden cubico y de orden c^n tienen un comportamiento similar, sin embargo se sigue comportando de peor manera la función de orden c^n , por esta razón se elige la función $O(n^3)$.

$O(n^3)$ vs $O(n!)$ 

Vemos en la gráfica que ambas funciones tienen un comportamiento similar, pero la función con orden factorial tiene mayor crecimiento, además como lo hemos dicho es el peor orden que podemos tener, de esta forma se elige la función de orden cuadrático.

$O(c^n)$ vs $O(n!)$ 

Del mismo modo que el caso anterior, tienen un caso muy similar, pero la de orden factorial sigue siendo la peor, por lo tanto, elegimos la función de orden c^n

Grafica comparativa de todos los órdenes de complejidad