Dibujo en blanco y negro

Descripción generada automáticamente con confianza baja**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

Joven con camiseta negra

Descripción generada automáticamente

**Índice**

[**Graficas en rango 0 < n < 100,000** 4](#_Toc84331619)

[**0(1) Complejidad constante** 4](#_Toc84331620)

[**0(log(n)) Complejidad logarítmica** 5](#_Toc84331621)

[**0(n) Complejidad lineal** 6](#_Toc84331622)

[**0(nlog(n)) Complejidad “n log n”** 7](#_Toc84331623)

[**0(n2) Complejidad cuadrática** 8](#_Toc84331624)

[**0(n3) Complejidad cubica** 9](#_Toc84331625)

[**0(cn);c > 1 Complejidad exponencial** 10](#_Toc84331626)

[**0(n!) Complejidad factorial** 10](#_Toc84331627)

[**Graficas por pares en rango 0 < n < 100** 11](#_Toc84331628)

[**O(1) vs O(log(n))** 11](#_Toc84331629)

[**O(1) vs O(n)** 12](#_Toc84331630)

[**O(1) vs O(nlog(n))** 13](#_Toc84331631)

[**O(1) vs O(n2)** 14](#_Toc84331632)

[**O(1) vs O(n3)** 15](#_Toc84331633)

[**O(1) vs O(cn)** 16](#_Toc84331634)

[**O(1) vs O(n!)** 17](#_Toc84331635)

[**O(log(n)) vs O(n)** 18](#_Toc84331636)

[**O(log(n)) vs O(nlog(n))** 19](#_Toc84331637)

[**O(log(n)) vs O(n2)** 20](#_Toc84331638)

[**O(log(n)) vs O(n3)** 21](#_Toc84331639)

[**O(log(n)) vs O(cn)** 22](#_Toc84331640)

[**O(log(n)) vs O(n!)** 23](#_Toc84331641)

[**O(n) vs O(nlog(n))** 24](#_Toc84331642)

[**O(n) vs O(n2)** 25](#_Toc84331643)

[**O(n) vs O(n3)** 26](#_Toc84331644)

[**O(n) vs O(cn)** 27](#_Toc84331645)

[**O(n) vs O(n!)** 28](#_Toc84331646)

[**O(nlog(n)) vs O(n2)** 29](#_Toc84331647)

[**O(nlog(n)) vs O(n3)** 30](#_Toc84331648)

[**O(nlog(n)) vs O(cn)** 31](#_Toc84331649)

[**O(nlog(n)) vs O(n!)** 32](#_Toc84331650)

[**O(n2) vs O(n3)** 33](#_Toc84331651)

[**O(n2) vs O(cn)** 34](#_Toc84331652)

[**O(n2) vs O(n!)** 35](#_Toc84331653)

[**O(n3) vs O(cn)** 36](#_Toc84331654)

[**O(n3) vs O(n!)** 37](#_Toc84331655)

[**O(cn) vs O(n!)** 38](#_Toc84331656)

[**Grafica comparativa de todos los órdenes de complejidad** 39](#_Toc84331657)

# **Graficas 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.

## **0(1) Complejidad constante**

Gráfico

Descripción generada automáticamente con confianza media

## **0(log(n)) Complejidad logarítmica**

Gráfico

Descripción generada automáticamente

## **0(n) Complejidad lineal**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## **0(nlog(n)) Complejidad “n log n”**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## **0(n2) Complejidad cuadrática**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## **0(n3) Complejidad cubica**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## **0(cn);c > 1 Complejidad exponencial**

Gráfico

Descripción generada automáticamente

## **0(n!) Complejidad factorial**

Tabla

Descripción generada automáticamente con confianza media

# **Graficas por pares en rango 0 < n < 100**

## **O(1) vs O(log(n))**

Gráfico

Descripción generada automáticamente

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)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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

## **O(1) vs O(nlog(n))**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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(n2)**

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

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(n3)**

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

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

## **O(1) vs O(cn)**

Gráfico

Descripción generada automáticamente

El orden cn 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!)**

Gráfico

Descripción generada automáticamente

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)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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(nlog(n))**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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 nlogn, siendo la de orden logn mejor.

## **O(log(n)) vs O(n2)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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 logn, por esa razón elegiría la de O(log(n)).

## **O(log(n)) vs O(n3)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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(cn)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Aquí como sucede en el caso anterior, la funcion de orden cn 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 funcion de orden constante 1, no es así pero da la impresión, por lo tanto elegiria esta ultima de orden logn.

## **O(log(n)) vs O(n!)**

Gráfico

Descripción generada automáticamente

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 logn.

## **O(n) vs O(nlog(n))**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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

## **O(n) vs O(n2)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

En esta comparación escogería la de orden n sobre la de orden n2 pues la diferencia se ve notoria en la gráfica al punto en que la de orden n parece tener un comportamiento constate.

## **O(n) vs O(n3)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Como pasa en el anterior ejemplos, en esta comparación la de orden n parece tener un comportamiento lineal respecto a la de orden n3, por lo tanto escogeria la de orden n.

## **O(n) vs O(cn)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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

## **O(n) vs O(n!)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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(nlog(n)) vs O(n2)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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

## **O(nlog(n)) vs O(n3)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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(nlog(n)) vs O(cn)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Podemos ver que el comportamiento del orden cn es de los peores casos que se puede dar, vemos en la gráfica que la función de orden nlogn esta muy por debajo, por lo tanto escogería esta.

## **O(nlog(n)) vs O(n!)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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 nlogn.

## **O(n2) vs O(n3)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Podemos pensar que la diferencia entre n2 y n3 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(n2) vs O(cn)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

De igual forma que en el apsado 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 cn, eligiendo de esta forma la funcion de orden cuadratico.

## **O(n2) vs O(n!)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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(n3) vs O(cn)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Podemos ver que la funcion de orden cubico y de orden cn tienen un comportamiento similar, sin embargo se sigue comportando de peor manera la funcion de orden cn, por esta razón se elige la función O(n3).

## **O(n3) vs O(n!)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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(cn) vs O(n!)**

Gráfico, Gráfico de líneas

Descripción generada automáticamente

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

# **Grafica comparativa de todos los órdenes de complejidad**

