**UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC- CTC**

**DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**PROJETO E ANÁLISE DE ALGORITMO**

**Prof. Alexandre Gonçalves Silva**

**Aluno: Osmar de Oliveira Braz Junior**

# 

# Questão 2

**2.** Em relação a crescimento de funções, pede-se:

(a) Para cada função *f*(*n*) e tempo *t* na tabela seguinte, determine o maior tamanho n de um problema que pode ser resolvido no tempo *t*, assumindo que o algoritmo para resolver o problema leve tempo *f*(*n*) nanosegundos (10-9 segundos).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tempo t  f(n)** nanosegundos | **1 Segundo** | **1 Minuto** | **1 Hora** | **1 Dia** | **1 Mês** | **1 Ano** | **1 Século** |
| **lg n \*** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **n** |  |  |  |  |  |  |  |
| **n lg n** |  |  |  |  |  |  |  |
| **n2** |  |  |  |  |  |  |  |
| **n3** |  |  |  |  |  |  |  |
| **2n** |  |  |  |  |  |  |  |
| **n!** |  |  |  |  |  |  |  |

R.:

Note que todos os tempos da coluna f(n) estão expressos em nanossegundos, ou seja, na ordem de .

Reescrevendo os tempos a serem calculados em função de segundos:

1 minuto = 60 segundos

1 hora = 60 minutos = 60 \* 60 segundos = 3.600 segundos

1 dia = 24 horas = 24 \* 3600 segundos = 86.400 segundos

Suponha 1 mês = 30 dias = 30 \* 86.400 segundos = 2.592.000 segundos

Suponha 1 ano = 365 dias = 365 \* 86.400 segundos = 31.536.000 segundos

1 século = 100 anos = 100 \* 31.536.000 segundos = 3.153.600.000 segundos

**Dados do Problema:**

**f(n)** Expressa o tempo de execução de um algoritmo em nanossegundos (10-9);

**Tempo** Representa o intervalo da iteração de f(n);

**O que se pede:**

Calcular o número de vezes que a função f(n) pode ser executada para cada unidade de tempo sugerida.

**Cálculo:**

Para encontrar o valor pedido, é necessário testar n a partir de 1, incrementando n (valores inteiros) em uma unidade até que f(n) iguale ou supere o Tempo em questão.

**Cálculos para as funções no tempo**

**=> lg n**

**Se log n \* segundo é <= ...**

**1 segundo:**

//Muda o sinal de -9 para positivo e multiplica por 10^9

// Mudar a base do log n de 2 para 10

//Multiplicar log 10 \* 10^9

// Definição =x equivale a=

//Eleve 2 ao expoente 10^9

//Funções exponencial e logarítmica são inversas

Logo, n =

**1 segundo:**

**1 minuto:** =

**1 hora:** =

**1 dia:** =

**1 mês:**=

**1 ano:** =

**1 século:** =

**=>**

**Se \* segundo é <= ...**

**1 segundo:**

//Eleva a ^2

//Elimina a raiz quadrada

Logo, n =

**1 minuto:**

//Eleva a ^2

//Adequa as potências

//Elimina a raiz quadrada

Logo, n =

**1 hora:**

//Eleva a ^2

//Adequa as potências

//Elimina a raiz quadrada

Logo, n =

**1 dia:**

//Eleva a ^2

//Adequa as potências

//Elimina a raiz quadrada

Logo, n =

**1 mês:**

//Eleva a ^2

//Adequa as potências

//Elimina a raiz quadrada

Logo, n =

**1 ano:**

//Eleva a ^2

//Adequa as potências

//Elimina a raiz quadrada

Logo, n =

**1 século: 3.153.600.000**

//Eleva a ^2

//Adequa as potências

//Elimina a raiz quadrada

Logo, n =

**1 segundo:**

**1 minuto:**

**1 hora:**

**1 dia:**

**1 mês:**

**1 ano:**

**1 século:**

**=>**

**Se n = segundo é <= ...**

**1 segundo:**

\*1

Logo, n =

**1 segundo:**

**1 minuto:** =

**1 hora:** =

**1 dia:** =

**1 mês:** =

**1 ano:** =

**1 século:** =

**=> n lg n**

**n=**

Testar os valores até o valor máximo de n que atenda a condição

Logo, n =

Desenvolvido o programa abaixo para calcular os valores para os tempos

public class NLogN{

//Realiza a mudanca de base de valor pela base

public static double log(double valor, double base) {

//Math.log logaritmo natural na base e

return Math.log(valor) / Math.log(base);

}

public static void main(String args[]){

//Tempo a serem verificados

double tempo[]= {1,60,3600,86400,2592000,31536000,3153600000.0};

//Percorre os tempos

for(int i = 0;i<tempo.length;i++){

//Para n lg n

double n = 1;

long passo = 1;

while ((n \* log(n,2)) <= Math.pow(10,9)\*tempo[i]){

n = n + passo;

//Acelerar o passo

if (n>100000000){

passo = passo + 10;

}

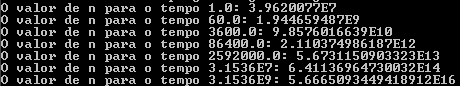
}

System.out.println("O valor de n para o tempo "+tempo[i]+": "+(n-1));

} }

}

Resultados da execução do algoritmo:



**1 segundo:**

**1 minuto:** 1**,**944659\*

**1 hora:** 9,857601\*

**1 dia:** 2,110374\*

**1 mês:** 5,673115\*

**1 ano:** 6,41136932\*

**1 século:** 5,666509\*

=> **n2**

**Se n2 \* segundo é <= ...**

**1 segundo:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz quadrada para os termos

//Isole 10^4 da raiz quadrada

Logo, n =

**1 minuto:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz quadrada para os termos

//Isole 10^5 da raiz quadrada

Logo, n =

**1 hora:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz quadrada para os termos

//Isole 10^6 da raiz quadrada

Logo, n =

**1 dia:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz quadrada para os termos

//Isole 10^6 da raiz quadrada

Logo, n =

**1 mês:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz quadrada para os termos

//Isole 10^7 da raiz quadrada

Logo, n =

**1 ano:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz quadrada para os termos

//Isole 10^8 da raiz quadrada

Logo, n =

**1 século:**

.000 //Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz quadrada para os termos

//Isole 10^9 da raiz quadrada

Logo, n =

**1 segundo:**

**1 minuto:**

**1 hora**

**1 dia:**

**1 mês:**

**1 ano:**

**1 século:**

=> **n3**

**Se n3 \* segundo é <= ...**

**1 segundo:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz cúbica para os termos

//Calcule a raiz cúbica

Logo, n =

**1 minuto:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz cúbica para os termos

//Isole 10^3 da raiz cúbica

//Calcule a raiz cúbica

Logo, n =

**1 hora:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz cúbica para os termos

//Isole 10^3 da raiz cúbica

//Calcule a raiz cúbica

Logo, n =

**1 dia:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz cúbica para os termos

//Isole 10^3 da raiz cúbica

//Calcule a raiz cúbica

Logo, n =

**1 mês:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz cúbica para os termos

//Organize as potências

//Isole 10^5 da raiz cúbica

//Calcule a raiz cúbica

Logo, n =

**1 ano:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz cúbica para os termos

//Organize as potências

//Isole 10^5 da raiz cúbica

//Calcule a raiz cúbica

Logo, n =

**1 século:**

//Muda o sinal de -9 para positivo

//Multiplica por 10^9

//Raiz cúbica para os termos

//Organize as potências

//Isole 10^6 da raiz cúbica

//Calcule a raiz cúbica

Logo, n =

**1 segundo:**

**1 minuto:**

**1 hora:**

**1 dia:**

**1 mês:**

**1 ano:**

**1 século:**

=> **2n**

**Se 2n \* segundo é <= ...**

**1 segundo:**

//Muda o sinal de -9 para positivo

// Definição equivale

//Mudar a base do log para 10

//

//

Logo, n =

**1 segundo:**

**1 minuto:**

**1 hora:**

**1 dia:**

**1 mês:**

**1 ano:**

**1 século:**

=> **n!**

Não existe função inversa de n!. Existe a aproximação de Stirling

Desenvolvido o programa abaixo para calcular os valores para os tempos

public class FatorialN{

//Calcula o fatorial de valor

public static double fatorial(double valor) {

double fat=1;

for(double i=1;i<=valor;i++){

fat = fat \* i;

}

return fat;

}

public static void main(String args[]){

//Tempo a serem verificados

double tempo[]= {1,60,3600,86400,2592000,31536000,3153600000.0};

//Percorre os tempos

for(int i = 0;i<tempo.length;i++){

//Para n lg n

double n = 1;

while (fatorial(n) <= Math.pow(10,9)\*tempo[i]){

n = n + 1;

}

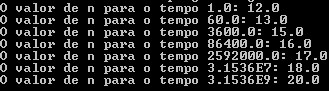
System.out.println("O valor de n para o tempo "+tempo[i]+": "+(n-1));

}

}

}

Resultados da execução do algoritmo:



**1 segundo:**

**1 minuto:**

**1 hora:**

**1 dia:** 16

**1 mês:**

**1 ano:**

**1 século:**

Tabela com os valores preenchido

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tempo t**  **f(n)=** | **1 Segundo** | **1 Minuto \* 60** | **1 Hora**  **\*3600** | **1 Dia**  **\*86400** | **1 Mês**  **\*2592000** | **1 Ano**  **\*31104000** | **1 Século \*3110400000** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| n |  |  |  |  |  |  |  |
| n lg n |  | 1**,**944659  \* | 9,857601  \* | 2,110374  \* | 5,673115  \* | 6,411369  \* | 5,666509  \* |
| n2 |  |  |  |  |  |  |  |
| n3 |  |  |  |  |  |  |  |
| 2n | 29,89735285 | 35,80424345 | 41,71113405 | 46,29609655 | 51,20298714 | 54,78794964 | 61,43180583 |
| n! | 12 | 13 | 15 | 16 | 17 | 18 | 20 |

(b) Considerando a seguinte lista de funções, ordene-as (da menor para a maior) em relação a taxa de crescimento, de modo que, se a função *g*(*n*) segue imediatamente a função *f*(*n*) em sua lista, então deveria ser o caso de *f*(*n*) ϵ O(*g*(*n*)).

*f*(*n*)=

*f*(*n*)=

*f*(*n*)=

*f*(*n*)=

*f*(*n*)=

*f*(*n*)=

R.:

As funções foram colocadas em ordem crescente levando em consideração a sua taxa de crescimento em relação a n.

*1o- f*(*n*)=

*2o- f*(*n*)=

*3o- f*(*n*)=

*4o- f*(*n*)=

*5o- f*(*n*)=

*6o- f*(*n*)=

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **N** | **f1(n) = n2,5** | **f2(n) = √2n** | **f3(n) = n + 10** | **f4(n) = 10n** | **f5(n) = 100n** | **f6(n) = n2 log2 n** |
| **1** | 1,00 | 1,41 | 11,00 | 1,00E+01 | 1,00E+02 | 0,00 |
| **2** | 5,66 | 2,00 | 12,00 | 1,00E+02 | 1,00E+04 | 4,00 |
| **3** | 15,59 | 2,45 | 13,00 | 1,00E+03 | 1,00E+06 | 14,26 |
| **4** | 32,00 | 2,83 | 14,00 | 1,00E+04 | 1,00E+08 | 32,00 |
| **5** | 55,90 | 3,16 | 15,00 | 1,00E+05 | 1,00E+10 | 58,05 |
| **6** | 88,18 | 3,46 | 16,00 | 1,00E+06 | 1,00E+12 | 93,06 |
| **7** | 129,64 | 3,74 | 17,00 | 1,00E+07 | 1,00E+14 | 137,56 |
| **8** | 181,02 | 4,00 | 18,00 | 1,00E+08 | 1,00E+16 | 192,00 |
| **9** | 243,00 | 4,24 | 19,00 | 1,00E+09 | 1,00E+18 | 256,76 |
| **10** | 316,23 | 4,47 | 20,00 | 1,00E+10 | 1,00E+20 | 332,19 |
| **11** | 401,31 | 4,69 | 21,00 | 1,00E+11 | 1,00E+22 | 418,59 |
| **12** | 498,83 | 4,90 | 22,00 | 1,00E+12 | 1,00E+24 | 516,23 |
| **13** | 609,34 | 5,10 | 23,00 | 1,00E+13 | 1,00E+26 | 625,37 |
| **14** | 733,36 | 5,29 | 24,00 | 1,00E+14 | 1,00E+28 | 746,24 |
| **15** | 871,42 | 5,48 | 25,00 | 1,00E+15 | 1,00E+30 | 879,05 |
| **16** | 1024,00 | 5,66 | 26,00 | 1,00E+16 | 1,00E+32 | 1024,00 |
| **17** | 1191,58 | 5,83 | 27,00 | 1,00E+17 | 1,00E+34 | 1181,28 |
| **18** | 1374,62 | 6,00 | 28,00 | 1,00E+18 | 1,00E+36 | 1351,06 |
| **19** | 1573,56 | 6,16 | 29,00 | 1,00E+19 | 1,00E+38 | 1533,50 |
| **20** | 1788,85 | 6,32 | 30,00 | 1,00E+20 | 1,00E+40 | 1728,77 |
| **21** | 2020,92 | 6,48 | 31,00 | 1,00E+21 | 1,00E+42 | 1937,01 |
| **22** | 2270,16 | 6,63 | 32,00 | 1,00E+22 | 1,00E+44 | 2158,36 |
| **23** | 2536,99 | 6,78 | 33,00 | 1,00E+23 | 1,00E+46 | 2392,96 |
| **24** | 2821,81 | 6,93 | 34,00 | 1,00E+24 | 1,00E+48 | 2640,94 |
| **25** | 3125,00 | 7,07 | 35,00 | 1,00E+25 | 1,00E+50 | 2902,41 |
| **26** | 3446,94 | 7,21 | 36,00 | 1,00E+26 | 1,00E+52 | 3177,50 |
| **27** | 3788,00 | 7,35 | 37,00 | 1,00E+27 | 1,00E+54 | 3466,31 |
| **28** | 4148,54 | 7,48 | 38,00 | 1,00E+28 | 1,00E+56 | 3768,97 |
| **29** | 4528,92 | 7,62 | 39,00 | 1,00E+29 | 1,00E+58 | 4085,56 |
| **30** | 4929,50 | 7,75 | 40,00 | 1,00E+30 | 1,00E+60 | 4416,20 |
| **31** | 5350,62 | 7,87 | 41,00 | 1,00E+31 | 1,00E+62 | 4760,98 |
| **32** | 5792,62 | 8,00 | 42,00 | 1,00E+32 | 1,00E+64 | 5120,00 |
| **33** | 6255,83 | 8,12 | 43,00 | 1,00E+33 | 1,00E+66 | 5493,35 |
| **34** | 6740,58 | 8,25 | 44,00 | 1,00E+34 | 1,00E+68 | 5881,11 |
| **35** | 7247,20 | 8,37 | 45,00 | 1,00E+35 | 1,00E+70 | 6283,37 |
| **36** | 7776,00 | 8,49 | 46,00 | 1,00E+36 | 1,00E+72 | 6700,22 |
| **37** | 8327,30 | 8,60 | 47,00 | 1,00E+37 | 1,00E+74 | 7131,74 |
| **38** | 8901,41 | 8,72 | 48,00 | 1,00E+38 | 1,00E+76 | 7578,01 |
| **39** | 9498,64 | 8,83 | 49,00 | 1,00E+39 | 1,00E+78 | 8039,10 |
| **40** | 10119,29 | 8,94 | 50,00 | 1,00E+40 | 1,00E+80 | 8515,08 |
| **41** | 10763,65 | 9,06 | 51,00 | 1,00E+41 | 1,00E+82 | 9006,04 |
| **42** | 11432,03 | 9,17 | 52,00 | 1,00E+42 | 1,00E+84 | 9512,05 |
| **43** | 12124,70 | 9,27 | 53,00 | 1,00E+43 | 1,00E+86 | 10033,16 |
| **44** | 12841,97 | 9,38 | 54,00 | 1,00E+44 | 1,00E+88 | 10569,46 |
| **45** | 13584,11 | 9,49 | 55,00 | 1,00E+45 | 1,00E+90 | 11121,00 |
| **46** | 14351,41 | 9,59 | 56,00 | 1,00E+46 | 1,00E+92 | 11687,86 |
| **47** | 15144,14 | 9,70 | 57,00 | 1,00E+47 | 1,00E+94 | 12270,09 |
| **48** | 15962,58 | 9,80 | 58,00 | 1,00E+48 | 1,00E+96 | 12867,75 |
| **49** | 16807,00 | 9,90 | 59,00 | 1,00E+49 | 1,00E+98 | 13480,92 |
| **50** | 17677,67 | 10,00 | 60,00 | 1,00E+50 | 1,00E+100 | 14109,64 |

As funções f1, f2, f3 e f6 agrupadas para a comparação pois pela tabela possuem comportamento assintótico em uma mesma escala.

As funções f4 e f5 foram coladas separadas pois possuem um comportamento assintótico muito maior que as outras funções. Neste caso f5 tem crescimento maior.

(c) Em cada um dos itens abaixo, indique se f(n) ϵ O(g(n)) ou f(n) ϵ Θ(g(n)) ou f(n) ϵ Ω(g(n)).

|  |  |  |  |
| --- | --- | --- | --- |
|  | *f*(*n*) | *g*(*n*) | Resultado |
| (a) |  |  | *f*(*n*) ϵ Θ(*g*(*n*)) |
| (b) |  |  | *f*(*n*) ϵ O(*g*(*n*)) |
| (c) |  |  | *f*(*n*) ϵ Θ(*g*(*n*)) |
| (d) |  |  | *f*(*n*) ϵ Θ(*g*(*n*)) |
| (e) |  |  | *f*(*n*) ϵ Θ(*g*(*n*)) |
| (f) |  |  | *f*(*n*) ϵ Θ(g(n)) |
| (g) |  |  | *f*(*n*) ϵ O(g(n)) |
| (h) |  |  | *f*(*n*) ϵ O(g(n)) |
| (i) |  |  | *f*(*n*) ϵ O(*g*(*n*)) |
| (j) |  |  | *f*(*n*) ϵ Ω(*g*(*n*)) |
| (k) |  |  | *f*(*n*) ϵ O(g(n)) |
| (l) |  |  | *f*(*n*) ϵ O(*g*(*n*)) |
| (m) |  |  | *f*(*n*) ϵ O(*g*(*n*)) |
| (n) |  |  | *f*(*n*) ϵ Ω(g(n)) |
| (o) |  |  | *f*(*n*) ϵ Ω(*g*(*n*)) |

Para resolvermos esta questão, é necessário se ater às seguintes regras enunciadas no Cormen (3ª Edição) e resumidas abaixo:

*Se , então f(n) ϵ Ω(g(n))*

*Se , então f(n) ϵ O(g(n))*

*Se então f(n) ϵ Θ(g(n))*

Onde n é uma entrada positiva.

Note que a base para o cálculo de c, tanto para *Ω* (g(n)), quanto para *O*(g(n)), parte do cálculo de . Diferindo apenas o teste a ser feito, conforme a seguir:

Se ), logo *f(n) ϵ Ω(g(n));*

Se , logo *f(n) ϵ O(g(n))*

A fim de simplificarmos os cálculos, procederemos inicialmente com a razão entre as funções, respeitando a limitação de c em comum em ambos os casos, ou seja, c>0 e então iniciaremos a análise específica de caso.

Observe que, para c maior do que zero, numerador e denominador de necessariamente deverão possuir o mesmo sinal, conforme tabela:

|  |  |  |
| --- | --- | --- |
| **Produto** | **+** | **-** |
| **+** | **+** | **-** |
| **-** | **-** | **+** |

Note, portanto, que, se o grau de n na razão for:

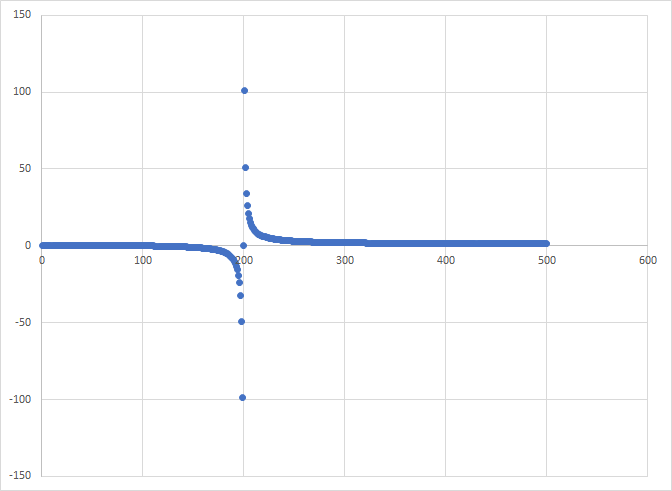
* Positivo, então *Ω(g(n));*pois f(n)>cg(n), ainda que para um c positivo gigantesco;
* Negativo, então *O(g(n));*pois f(n)<cg(n), ainda que para um c positivo mínimo;
* Nulo, então *Θ(g(n));*pois f(n)>cg(n), ainda que para um c positivo gigantesco, E f(n)<cg(n), ainda que para um c positivo mínimo.

**(a) f(n)=n-100;  g(n)=n-200**

Calculando o grau da razão

=

A diferença entre os termos de maior grau na razão é 0, logo *f(n) ϵ Ω (g(n)) f(n) ϵ O (g(n))* portanto***f(n) ϵ Θ (g(n))***

**

No gráfico é apresentado o comportamento da razão, que é negativo (e descontínuo em n=200) para valores de n<200.

Para que c seja positivo, a razão também deverá ser positiva, logo, n >200.

Então c >0 ∀n > 200

***f(n) ϵ Ω (g(n)) ?***

supondo c=2, logo n0 ...

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição  *.*

Ou seja: f(n) está em *Ω* (g) para c=2 e n=300

0 <= 2\*(300-200) <= 300-100

0 <= 2\*(100) <= 200

0 <= 200 <= 200 = Verdadeiro

Logo , *f(n) ϵ Ω (g(n))*

***f(n) ϵ O (g(n)) ?***

supondo c=2, logo n0 ...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=2 e n=300

0 <= 300-100 <= 2\*(300-200)

0 <= 200 <= 2\*(100)

0 <= 200 <= 200 = Verdadeiro

Logo, *f(n) ϵ O (g(n))*

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) ϵ Ω (g(n)) e f(n) ϵ O (g(n))*então***f(n) ϵ Θ (g(n))***

Por exemplo, existem c1 e c2 (c1=c2=2, no=300) de modo que a inequação

é verdadeira

Logo ***f(n) ϵ Θ (g(n))***

**(b) f(n)=;  g(n)=**

Calculando grau da razão

= //Fatore

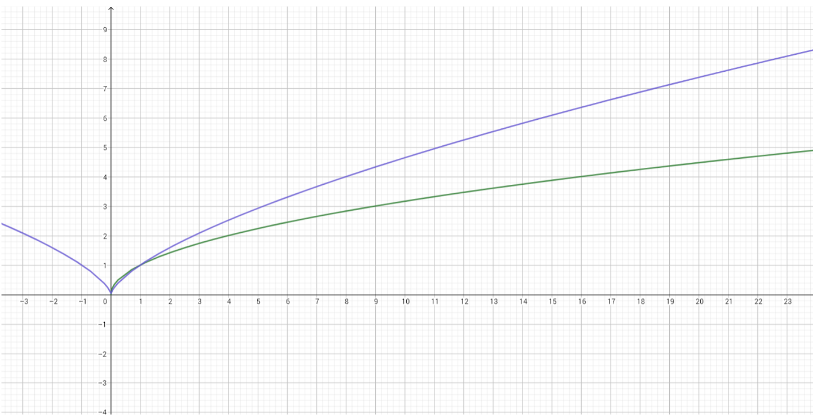
//MMC de 2 e 3

=

= //Cancele o fator comum

=

Comparando graficamente as funções f(n) e g(n) identificamos que g(n) possui um comportamento assintótico muito superior ao de f(n), de forma que a ordem de g(n) é superior.



No gráfico é apresentado o comportamento da razão, contínua e positiva, para todo n positivo. Para que c seja positivo, a razão também deverá ser positiva, logo, n >1.

c >0 ∀n > 1

***f(n) ϵ Ω (g(n)) ?***

supondo c=1, logo n0 ...

// Eleve cada lado da equação a potência 6/1

//Um elevado a qualquer potência é um

f(n) *Ω* (g(n)) porque não existe um c>1 e n que satisfaz a condição . Ou seja: f(n) não está em *Ω* (g) pois f(n) sempre será menor que g(n).

c=2 e n=1

0 <= 2\* <=

0 <= 2 <= 1

0 <= 2 <= 1 = Falso

Logo, *f(n) Ω (g(n))*

***f(n) ϵ O (g(n)) ?***

supondo c=1, logo n0 ...

// Eleve cada lado da equação a potência 6/1

//Um elevado a qualquer potência é um

**c=2**

// Eleve cada lado da equação a potência -6/1

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=1 e n=1

0 <= <= 1\*

0 <= 1 <= 1\*1

0 <= 1 <= 1 = Verdadeiro

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) Ω (g(n))* então ***f(n) Θ (g(n))***

Logo, ***f(n) ϵ O (g(n))***

**c) ;**

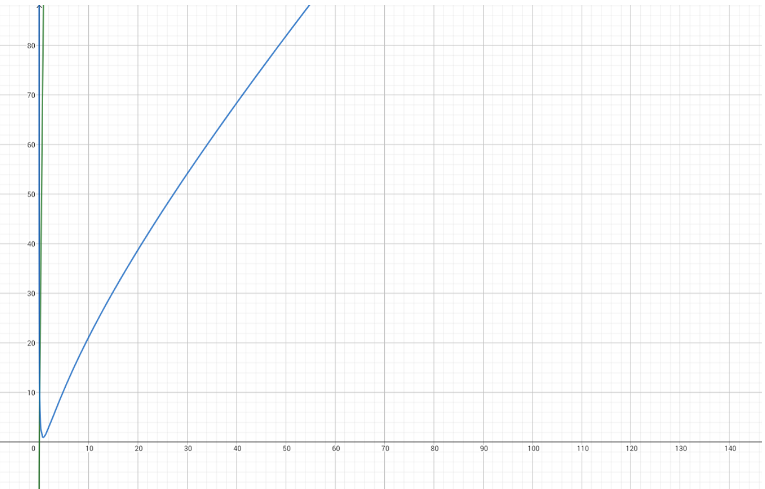
Calculando o grau da razão :

=

A diferença entre os termos de maior grau na razão é 0, logo *f(n) ϵ Ω (g(n)) f(n) ϵ O (g(n))* então***f(n) ϵ Θ (g(n)).***

Comparando graficamente as funções f(n) e g(n) observamos que o comportamento de f(n) é assintoticamente superior, porém, ambos encontram-se na mesma ordem, de modo que, se aplicássemos um fator a g(n), superaríamos f(n).

Logo, ***f(n) ϵ Θ (g(n)).***

**

No gráfico é apresentado o comportamento da razão, que é positiva para todo n positivo.

***f(n) ϵ Ω (g(n)) ?***

supondo c=1, logo n0=2...

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição  *.*

Ou seja: f(n) está em *Ω* (g) para c=1 e n=2

***f(n) ϵ O (g(n)) ?***

supondo c=128, logo n0=2...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=128 e no=2

Logo, ***f(n) ϵ O (g(n))***

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) ϵ Ω (g(n)) e f(n) ϵ O (g(n))* então ***f(n) ϵ Θ (g(n))***

Por exemplo, existem c1 e c2 (c1=1, c2=128, no=2) de modo que a inequação

é verdadeira

Logo ***f(n) ϵ Θ (g(n))***

**(d) f(n)=;  g(n)=**

Calculando o grau da razão

=

Note que f(n) tem ordem maior que g(n), logo *f(n) ϵ Θ (g(n))*

***f*(*n*) ϵ Ω(g(n))?**

supondo c=0,01, logo n0=8...

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição  *.*

Ou seja: f(n) está em *Ω* (g) para c=0,01 e n=8

***f*(*n*) ϵ O(g(n))?**

supondo c=1, logo n0=8...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=1 e no>=8

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) ϵ Ω (g(n)) e f(n) ϵ O (g(n))* então ***f(n) ϵ Θ (g(n))***

Por exemplo, existem c1 e c2 (c1=0,01, c2=1, no=8) de modo que a inequação

é verdadeira

Logo ***f(n) ϵ Θ (g(n))***

**(e) f(n)=;  g(n)=**

Calculando o grau da razão

=

=

=

Após manipularmos as funções originais, percebe-se claramente que estamos diante da mesma função, apenas variando a constante, que não interfere no comportamento da função para n suficientemente grande, *f(n) ϵ Θ (g(n)).*

***f*(*n*) ϵ Ω(g(n)) ?**

supondo c=0.8, logo n0=4...

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição  *.*

Ou seja: f(n) está em *Ω* (g) para c=0.8 e n=3

***f(n) ϵ O (g(n)) ?***

supondo c=1, logo n0=4...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=1 e no=4

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) ϵ Ω (g(n)) e f(n) ϵ O (g(n))* então ***f(n) ϵ Θ (g(n))***

Logo ***f(n) ϵ Θ (g(n))***

**(f) f(n)=;  g(n)=**

Calculando o grau da razão

=

=

=

A diferença entre os termos de maior grau na razão é 0, logo *f(n) ϵ Ω (g(n)) f(n) ϵ O (g(n))* então***f(n) ϵ Θ (g(n))*** *é verdadeira*

Logo ***f(n) ϵ Θ (g(n))***

***f(n) ϵ Ω (g(n)) ?***

supondo c=1, logo n0=16 ....

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição  *.*

Ou seja: f(n) está em *Ω* (g) para c=1 e n=16

***f(n) ϵ O (g(n)) ?***

supondo c=10, logo n0=16...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=10 e n=16

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) ϵ Ω (g(n)) e f(n) ϵ O (g(n))* então***f(n) ϵ Θ (g(n))***

Por exemplo, existem c1 e c2 (c1=1, c2=10, no=16) de modo que a inequação

é verdadeira

Logo ***f(n) ϵ Θ (g(n))***

**(g) f(n)=;  g(n)= *f*(*n*) ϵ Ω(g(n))**

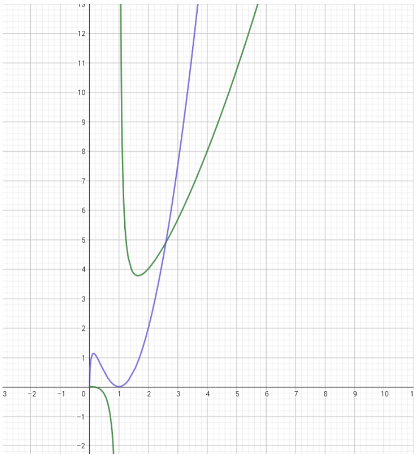
Calculando o grau da razão

=

=

=

Comparando graficamente as funções, identificamos que f(n) possui um comportamento assintótico muito superior ao de g(n), de forma que a ordem de f(n) é superior.



***f(n) ϵ*** *Ω* ***(g(n)) ?***

A função f(n) possui ordem superior a g(n), portanto *f(n) Ω (g(n))*

f(n) *Ω* (g(n))

***f(n) ϵ O (g(n)) ?***

supondo c=0.3 logo n0 =16, ...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=0.3 e no=16

Logo, ***f(n) ϵ O (g(n))***

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) Ω (g(n))* então ***f(n) Θ (g(n))***

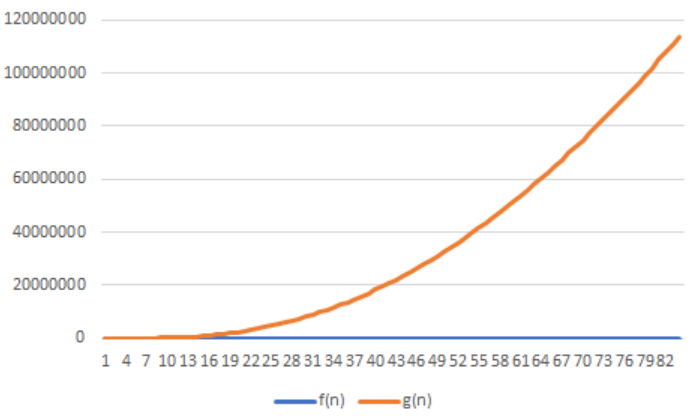
Logo, ***f(n) ϵ O (g(n))***

**(h) f(n)=;  g(n)=**

Calculando o grau da razão

=

Comparando graficamente as funções, identificamos que g(n)possui um comportamento assintótico muito superior ao de f(n), de forma que a ordem de g(n) é superior.



Portanto, ***f(n) ϵ O (g(n))***

***f(n) ϵ*** *Ω* ***(g(n)) ?***

A função f(n) possui ordem superior a g(n), portanto *f(n) Ω (g(n))*

***f(n) ϵ O (g(n)) ?***

supondo c=1 logo n0 =4, ...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=1 e n=4

Logo, ***f(n) ϵ O (g(n))***

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) Ω (g(n))* então ***f(n) Θ (g(n))***

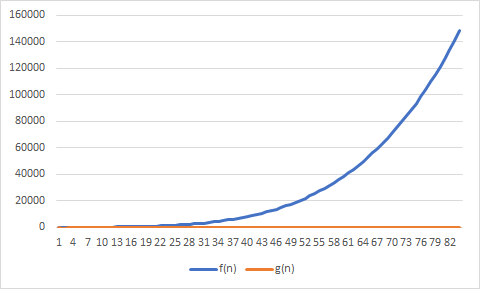
Logo, ***f(n) ϵ O (g(n))***

**(i) f(n)=;  g(n)=**

Calculando o grau da razão

=

Comparando graficamente as funções f(n)= e g(n)= identificamos que f(n)possui um comportamento assintótico muito superior ao de g(n), de forma que a ordem de f(n) é superior.

**

Portanto, ***f(n) ϵ O (g(n))***

***f(n) ϵ Ω (g(n)) ?***

A função g(n) tem ordem maior que f(n).

Portanto ***f(n)*** *Ω* ***(g(n))***

***f(n) ϵ O (g(n)) ?***

supondo c=1 logo n0 =2

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c =1 e n=2

Logo, ***f(n) ϵ O (g(n))***

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) Ω (g(n))* então ***f(n) Θ (g(n))***

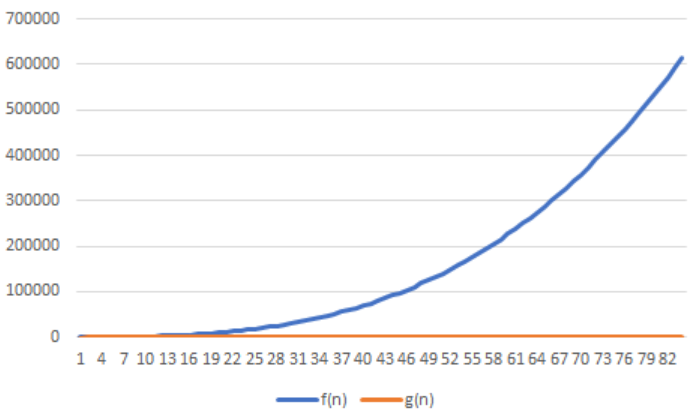
Logo, ***f(n) ϵ O (g(n))***

**(j) f(n)=;  g(n)=**

Calculando o grau da razão

=

Comparando graficamente as funções f(n) e g(n) identificamos que f(n)possui um comportamento assintótico muito superior ao de g(n), de forma que a ordem de f(n) é superior.



Portanto, ***f(n) ϵ*** *Ω* ***(g(n))***

***f(n) ϵ*** *Ω* ***(g(n)) ?***

supondo c=0.1 logo n0 =4...

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição  *.*

Ou seja: f(n) está em *Ω* (g) para c=0.1 e n=4

***f(n) ϵ O (g(n)) ?***

Como *f(n) O (g(n)) e f(n) ϵ Ω (g(n))*então***f(n) ϵ*** *Ω* ***(g(n))***

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) O (g(n))* então ***f(n) Θ (g(n))***

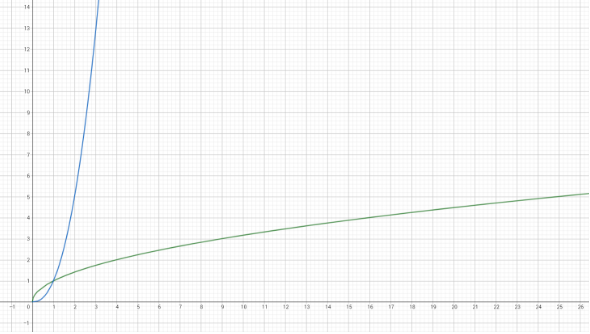
Logo, ***f(n) ϵ*** *Ω* ***(g(n))***

**(k) f(n)=;  g(n)=**

Calculando o grau da razão

=

Comparando graficamente as funções f(n) e g(n) identificamos que g(n)possui um comportamento assintótico muito superior ao de f(n), de forma que a ordem de g(n) é superior.



Portanto, g(n) tem ordem maior ***f(n) ϵ*** *O* ***(g(n))***

***f(n) ϵ*** *Ω* ***(g(n)) ?***

A função g(n) tem ordem maior que f(n).

Portanto ***f(n)*** *Ω* ***(g(n))***

***f(n) ϵ O (g(n)) ?***

supondo c=0.05 logo n0=16...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para =0.05 e n=16

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) Ω (g(n))* então ***f(n) Θ (g(n))***

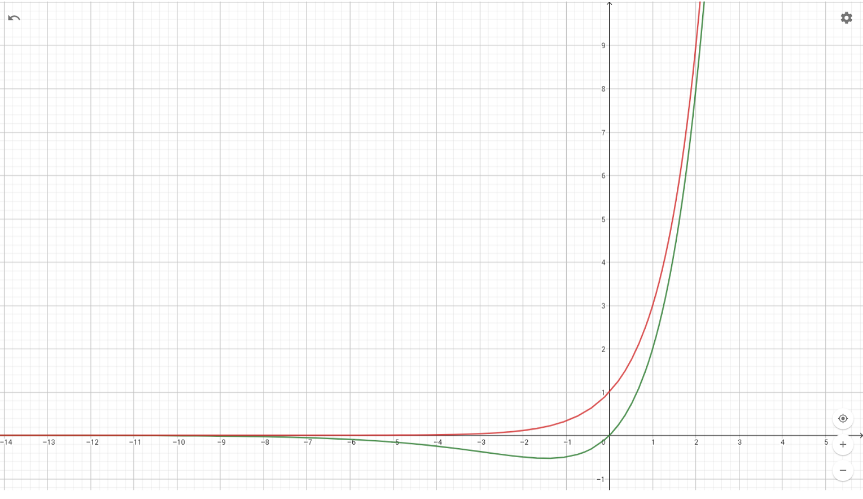
Logo, ***f(n) ϵ*** *O* ***(g(n))***

**(l) f(n)=;  g(n)=**

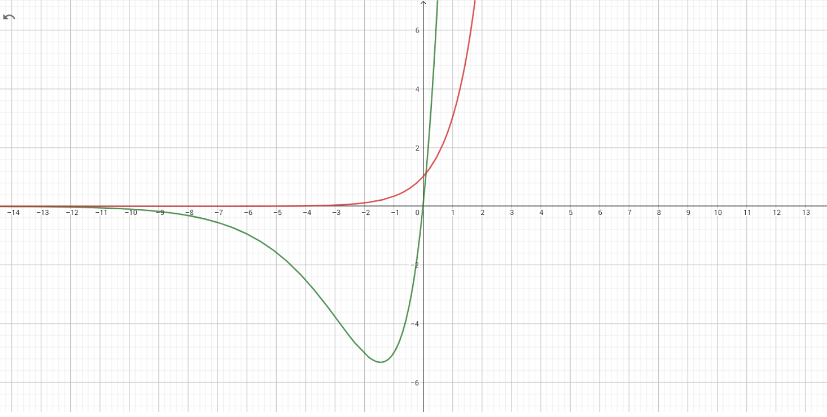
Calculando o grau da razão

=

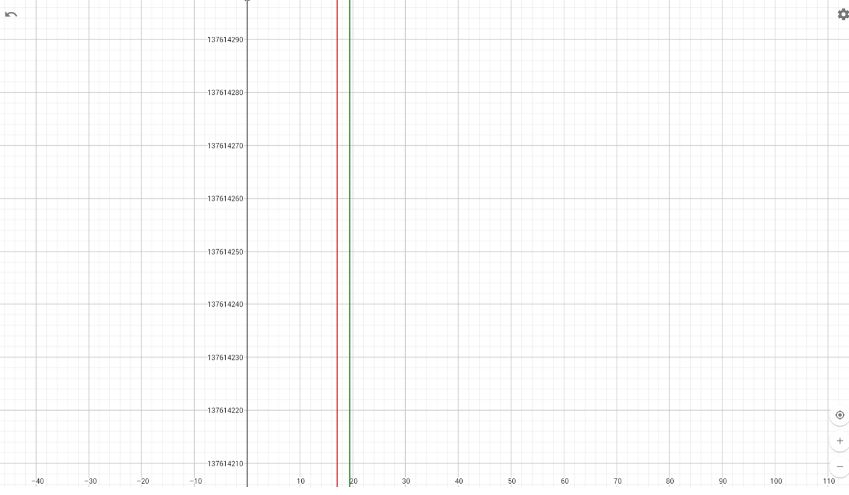
Comparando graficamente as funções f(n) e g(n) identificamos que g(n) possui um comportamento assintótico muito superior ao de f(n), de forma que a ordem de g(n) é superior.



estando algumas constantes *w|wf(n)*, na tentativa de superar g(n), provou-se infrutífera para n suficientemente grande. Observe a comparação entre *10f(n)* e *g(n)*.

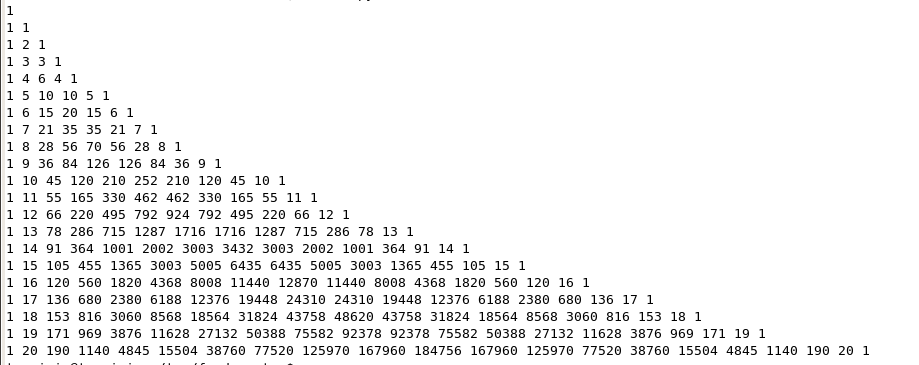


Com isto f(n) foi superado.



Através de uma pequena manipulação algébrica, podemos reescrever como um binômio a fim de melhor estimar o crescimento para n suficientemente grande em relação a , ou seja, .

O Triângulo de Pascal enuncia que os coeficientes de expansões binomiais podem ser calculados conforme triângulo abaixo.

**

Existe uma regra de formação onde os coeficientes da linha L determinam os coeficientes da linha L+1.

Exemplo:

Note que os coeficientes correspondem à segunda linha do Triângulo: 1, 2, 1.

Quando um binômio é da forma , elevado a uma potência, temos:

onde corresponde aos coeficientes da n-ésima linha.

Portanto, g(n) tem ordem maior que ***f(n) ϵ*** *O* ***(g(n))***

***f(n) ϵ*** *Ω* ***(g(n)) ?***

A função g(n) tem ordem maior que f(n).

Portanto ***f(n)*** *Ω* ***(g(n))***

***f(n) ϵ O (g(n)) ?***

supondo c=0.8 logo n0=4...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=0.8 e n=14

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) Ω (g(n))* então ***f(n) Θ (g(n))***

Logo, ***f(n) ϵ*** *O* ***(g(n))***

**(m) f(n)=;  g(n)= *f*(*n*) ϵ Θ(g(n))**

Calculando o grau da razão

=

=

= //Cancela 2^n

=

Existe uma proporção na razão, portanto *f(n) ϵ Θ (g(n)).*

***f*(*n*) ϵ Ω(g(n)) ?**

supondo c=0.1, logo n0=4...

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição *.*

Ou seja: f(n) está em *Ω* (g) para c=0.1 e n=4

***f(n) ϵ O (g(n)) ?***

supondo c=1, logo n0=4...

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição

Ou seja: f(n) está em O(g) para c=1 e no=4

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) ϵ Ω (g(n)) e f(n) ϵ O (g(n))* então ***f(n) ϵ Θ (g(n))***

Por exemplo, existem c1 e c2 (c1=0.1, c2=1, no=4) de modo que a inequação

é verdadeira

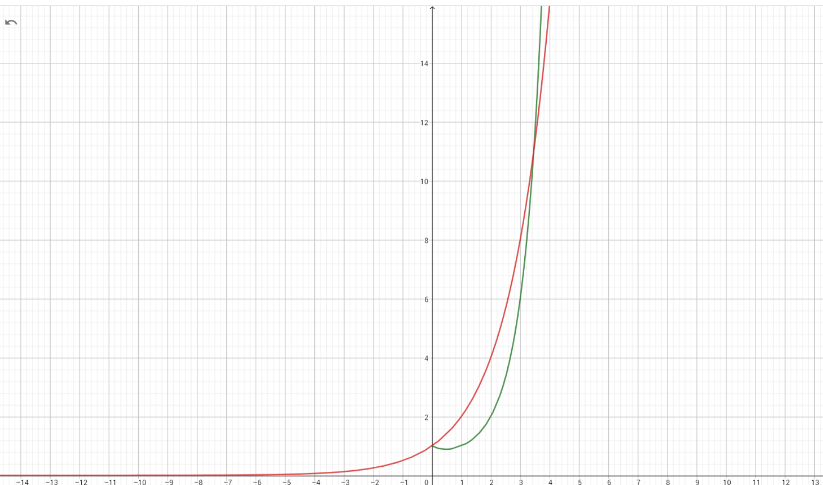
Logo ***f(n) ϵ Θ (g(n))***

**(n) f(n)=;  g(n)= *f*(*n*) ϵ Ω (g(n))**

Calculando o grau da razão

=

Comparando graficamente f(n)=n!(verde) e g(n)=2n(vermelho) conclui-se que g(n)terá um crescimento muito superior para n suficientemente grande.



De fato, não é difícil concluir que o crescimento de f(n)é muito maior, ainda que multipliquemos qualquer constante pois g(n)sempre dobrará seu valor enquanto a cada incremento de n, enquanto f(n) será n vezes maior.

***f(n) ϵ*** *Ω* ***(g(n)) ?***

supondo c=1 logo n0 =4...

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição  *.*

Ou seja: f(n) está em *Ω* (g) para c=1 e n=4

***f(n) ϵ O (g(n)) ?***

Como *f(n) O (g(n)) e f(n) ϵ Ω (g(n))*então***f(n) ϵ*** *Ω* ***(g(n))***

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) O (g(n))* então ***f(n) Θ (g(n))***

Logo, ***f(n) ϵ*** *Ω* ***(g(n))***

**(o) f(n)=;  g(n)=**

Calculando o grau da razão

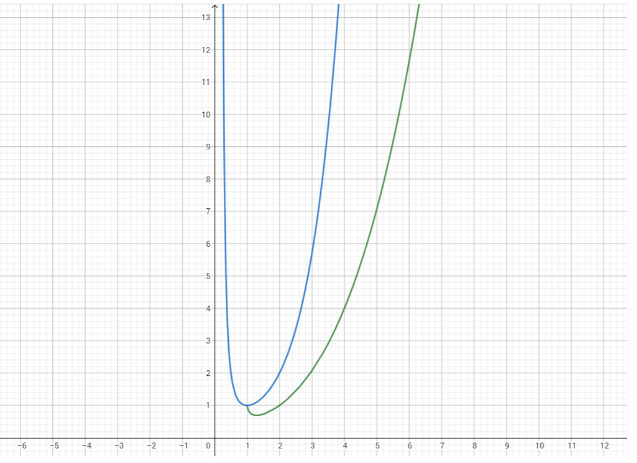
=

Suponha

=

=

Comparando graficamente f(n) e g(n) conclui-se que g(n)terá um crescimento mais acentuado superior para n suficientemente grande.



Além de g(n) apresenta um crescimento mais acentuado, também é notório que seu grau de crescimento é maior em função do expoente 2. Logo, f(n) ϵ *Ω* (g(n)).

Existe uma proporção na razão, portanto *f(n) ϵ Θ (g(n)).*

***f*(*n*) ϵ Ω(g(n)) ?**

supondo c=1, logo n0=8...

f(n) ϵ *Ω* (g(n)) porque existe um c e n que satisfaz a condição *.*

Ou seja: f(n) está em *Ω* (g) para c=1 e n=8

***f(n) ϵ O (g(n)) ?***

Como *f(n) O (g(n)) e f(n) ϵ Ω (g(n))*então***f(n) ϵ*** *Ω* ***(g(n))***

***f(n) ϵ Θ (g(n)) ?***

Como *f(n) O (g(n))* então ***f(n) Θ (g(n))***

Logo, ***f(n) ϵ*** *Ω* ***(g(n))***