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= 10^{-9}	**1 Segundo**	**1 Minuto**	**1 Hora**	**1 Dia**	**1 Mês**	**1 Ano**	**1 Século**
lg n *							
$\sqrt{n}$							
n							
n lg n							
n^2							
n^3							
2^n							
n!							

R.:

Note que todos os tempos da coluna f(n) estão expressos em nanossegundos, ou seja, na ordem de 10^{-9}.

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 * 10^{-9} segundo é <= ...

1 segundo:

$\frac{\log_2 n}{10^9} \leq 1$ //Muda o sinal de -9 para positivo e multiplica por 10^9

$\log_2 n \leq 10^9$ // Mudar a base do log n de 2 para 10

$\frac{\log_{10} n}{\log_{10} 2} \leq 10^9$ //Multiplicar log 10 * 10^9

$\log_{10} n \leq \log_{10} 2 * 10^9$// Definição$\log_b a$ =x equivale a=b^x

$n \leq 10^{\log_{10} 2 * 10^9}$//Eleve 2 ao expoente 10^9

$n \leq 10^{\log_{10} 2^{10^9}}$ //Funções exponencial e logarítmica são inversas

$n \leq 2^{10^9}$

Logo, n = 2^{10^9}

1 segundo: 2^{10^9}

1 minuto: 2^{60*10^9}=$2^{6*10^{10}}$

1 hora: $2^{3600*10^9}$=$2^{3,600*10^{12}}$

1 dia: $2^{86400*10^9}$=$2^{8,6400*10^{13}}$

1 mês:$2^{2592000*10^9}$ =$2^{2,592000*10^{15}}$

1 ano: $2^{31536000*10^9}$ =$2^{3,1536000*10^{16}}$

1 século: $2^{31536000000*10^9}$ =$2^{3,1536000000*10^{18}}$

=>$\sqrt{n}$

Se $\sqrt{n}$ * 10^{-9} segundo é <= ...

1 segundo:

$\frac{\sqrt{n}}{10^9} \leq 1$

$\sqrt{n} \leq 10^9$//Eleva a ^2

$(\sqrt{n})^2 \leq (10^9)^2$ //Elimina a raiz quadrada

$n \leq 10^{18}$

Logo, n = 10^{18}

1 minuto:

$\frac{\sqrt{n}}{10^9} \leq 60$

$\sqrt{n} \leq 60 * 10^9$//Eleva a ^2

$(\sqrt{n})^2 \leq (6 * 10^{10})^2$ //Adequa as potências

$(\sqrt{n})^2 \leq (6 * 10^{10})^2$ //Elimina a raiz quadrada

$n \leq 3.6 * 10^{21}$

Logo, n = $3.6 * 10^{21}$

1 hora:

$\frac{\sqrt{n}}{10^9} \leq 3600$

$\sqrt{n} \leq 3600 * 10^9$//Eleva a ^2

$(\sqrt{n})^2 \leq (3600 * 10^9)^2$ //Adequa as potências

$(\sqrt{n})^2 \leq (3.6 * 10^{12})^2$ //Elimina a raiz quadrada

$n \leq 12.96 * 10^{24}$

$n \leq 1{,}296 * 10^{25}$

Logo, n = $1{,}296 * 10^{25}$

1 dia:
$\frac{\sqrt{\boldsymbol{n}}}{10^9} \leq 86400$
$\sqrt{\boldsymbol{n}} \leq \ 86400 * 10^9$//Eleva a ^2
$(\sqrt{\boldsymbol{n}})^2 \leq \ (86400 * 10^9)^2$ //Adequa as potências
$(\sqrt{\boldsymbol{n}})^2 \leq \ (8.64 * 10^{13})^2$ //Elimina a raiz quadrada
$\boldsymbol{n} \leq 74{,}6496 * 10^{26}$
$\boldsymbol{n} \leq 7{,}46496 * 10^{27}$
Logo, n = $7{,}46496 * 10^{27}$

1 mês:
$\frac{\sqrt{\boldsymbol{n}}}{10^9} \leq 2592000$
$\sqrt{\boldsymbol{n}} \leq \ 2592000 \ * 10^9$//Eleva a ^2
$(\sqrt{\boldsymbol{n}})^2 \leq \ (2592000 \ * 10^9)^2$ //Adequa as potências
$(\sqrt{\boldsymbol{n}})^2 \leq \ (2{,}592 \ * 10^{15})^2$ //Elimina a raiz quadrada
$\boldsymbol{n} \leq \ 6{,}718464 * 10^{30}$
Logo, n = $6{,}718464 * 10^{30}$

1 ano:
$\frac{\sqrt{\boldsymbol{n}}}{10^9} \leq 31536000$
$\sqrt{\boldsymbol{n}} \leq \ 31536000 * 10^9$//Eleva a ^2
$(\sqrt{\boldsymbol{n}})^2 \leq \ (31536000 * 10^9)^2$ //Adequa as potências
$(\sqrt{\boldsymbol{n}})^2 \leq \ (3{,}1536 \ * 10^{16})^2$ //Elimina a raiz quadrada
$\boldsymbol{n} \leq \ 9{,}94519296 * 10^{32}$
Logo, n = $9{,}94519296 * 10^{32}$

1 século: 3.153.600.000
$\frac{\sqrt{\boldsymbol{n}}}{10^9} \leq 3153600000$
$\sqrt{\boldsymbol{n}} \leq \ 3153600000 * 10^9$//Eleva a ^2
$(\sqrt{\boldsymbol{n}})^2 \leq \ (3153600000 * 10^9)^2$ //Adequa as potências
$(\sqrt{\boldsymbol{n}})^2 \leq \ (3{,}1536 \ * 10^{18})^2$ //Elimina a raiz quadrada
$\boldsymbol{n} \leq \ 9{,}94519296 * 10^{36}$
Logo, n = $9{,}94519296 * 10^{36}$

1 segundo: 10^{18}
1 minuto: $3.6 * 10^{21}$
1 hora: $1{,}296 * 10^{25}$
1 dia: $7{,}46496 * 10^{27}$
1 mês: $6{,}718464 * 10^{30}$
1 ano: $9{,}94519296 * 10^{32}$
1 século: $9{,}94519296 * 10^{38}$

=>$\boldsymbol{n}$
Se n = 10^{-9} segundo é <= ...
1 segundo:
$n = 10^9$ *1
Logo, n = 10^9

1 segundo: 10^9
1 minuto: $60 * 10^9$=$6 * 10^{10}$

1 hora: $3600 * 10^9$=$3,6 * 10^{12}$
1 dia: $86400 * 10^9$=$8,64 * 10^{13}$
1 mês: $2.592.000 * 10^9$=$2,592 * 10^{15}$
1 ano: $31.536.000 * 10^9$=$3,1536 * 10^{16}$
1 século: $3.153.600.000 * 10^9$=$3,1536 * 10^{18}$

=> n lg n
n=10^9
$n \log n \; * 10^{-9} \leq 1$
$\frac{\mathrm{n} \log_2 n}{10^9} \leq \; 1$
$\mathrm{n} \log_2 n \leq 10^9$
Testar os valores até o valor máximo de n que atenda a condição
Logo, n = $3,9620077 * 10^3 \leq 10^9$

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:

```
O valor de n para o tempo 1.0: 3.9620077E7
O valor de n para o tempo 60.0: 1.944659487E9
O valor de n para o tempo 3600.0: 9.8576016639E10
O valor de n para o tempo 86400.0: 2.110374986187E12
O valor de n para o tempo 2592000.0: 5.6731150903323E13
O valor de n para o tempo 3.1536E7: 6.41136964730032E14
O valor de n para o tempo 3.1536E9: 5.6665093449418912E16
```

1 segundo: $3,962000 * 10^7$
1 minuto: $1,944659*10^9$
1 hora: $9,857601*10^{10}$
1 dia: $2,110374*10^{12}$
1 mês: $5,673115*10^{13}$
1 ano: $6,41136932*10^{14}$

1 século: 5,666509*10^{16}

=> n²
Se n² * 10^{-9} segundo é <= ...
1 segundo:
$\frac{n^2}{10^9} \leq 1$ //Muda o sinal de -9 para positivo
$n^2 \leq 1 * 10^9$ //Multiplica por 10^9
$\sqrt{n^2} \leq \sqrt{10^9}$//Raiz quadrada para os termos
$n \leq \sqrt{10^9}$ //Isole 10^4 da raiz quadrada
$n \leq \sqrt{10 * 10^4}$
$n \leq 3{,}162277 * 10^4$
Logo, n = $3{,}162277 * 10^4$

1 minuto:
$\frac{n^2}{10^9} \leq 60$ //Muda o sinal de -9 para positivo
$n^2 \leq 60 * 10^9$ //Multiplica por 10^9
$\sqrt{n^2} \leq \sqrt{60 * 10^9}$//Raiz quadrada para os termos
$n \leq \sqrt{6 * 10^{10}}$ //Isole 10^5 da raiz quadrada
$n \leq \sqrt{6} * 10^5$
$n \leq 2{,}449489 * 10^5$
Logo, n = $2{,}449489 * 10^5$

1 hora:
$\frac{n^2}{10^9} \leq 3600$ //Muda o sinal de -9 para positivo
$n^2 \leq 3600 * 10^9$ //Multiplica por 10^9
$\sqrt{n^2} \leq \sqrt{3600 * 10^9}$//Raiz quadrada para os termos
$n \leq \sqrt{3{,}6 * 10^{12}}$ //Isole 10^6 da raiz quadrada
$n \leq \sqrt{6} * 10^6$
$n \leq 1{,}897366 * 10^6$
Logo, n = $1{,}897366 * 10^6$

1 dia:
$\frac{n^2}{10^9} \leq 86.400$//Muda o sinal de -9 para positivo
$n^2 \leq 86.400 * 10^9$ //Multiplica por 10^9
$\sqrt{n^2} \leq \sqrt{86.400 * 10^9}$//Raiz quadrada para os termos
$n \leq \sqrt{86{,}4 * 10^{12}}$ //Isole 10^6 da raiz quadrada
$n \leq \sqrt{86{,}4} * 10^6$
$n \leq 9{,}295160 * 10^6$
Logo, n = $9{,}295160 * 10^6$

1 mês:
$\frac{n^2}{10^9} \leq 2.592.000$//Muda o sinal de -9 para positivo
$n^2 \leq 2.592.000 * 10^9$ //Multiplica por 10^9
$\sqrt{n^2} \leq \sqrt{2.592.000 * 10^9}$//Raiz quadrada para os termos
$n \leq \sqrt{25{,}92 * 10^{14}}$ //Isole 10^7 da raiz quadrada
$n \leq \sqrt{25{,}92 * 10^7}$
$n \leq 5{,}091168 * 10^7$
Logo, n = $5{,}091168 * 10^7$

1 ano:
$\frac{n^2}{10^9} \leq 31.536.000$//Muda o sinal de -9 para positivo
$n^2 \leq 31.536.000 * 10^9$ //Multiplica por 10^9
$\sqrt{n^2} \leq \sqrt{31.536.000 * 10^9}$//Raiz quadrada para os termos
$n \leq \sqrt{3{,}1536 * 10^{16}}$ //Isole 10^8 da raiz quadrada
$n \leq \sqrt{3{,}1536 * 10^8}$
$n \leq 1{,}775837 * 10^8$
Logo, n = $1{,}775837 * 10^8$

1 século:
$\frac{n^2}{10^9} \leq 31.536.000.000$ //Muda o sinal de -9 para positivo
$n^2 \leq 31.536.000.000 * 10^9$ //Multiplica por 10^9
$\sqrt{n^2} \leq \sqrt{31.536.000.000 * 10^9}$//Raiz quadrada para os termos
$n \leq \sqrt{31{,}536 * 10^{18}}$ //Isole 10^9 da raiz quadrada
$n \leq \sqrt{31{,}536} * 10^9$
$n \leq 5{,}615692 * 10^9$
Logo, n = $5{,}615692 * 10^9$

1 segundo:$3{,}162277 * 10^4$
1 minuto: $2{,}449489 * 10^5$
1 hora$1{,}897366 * 10^6$
1 dia: $9{,}295160 * 10^6$
1 mês: $5{,}091168 * 10^7$
1 ano: $1{,}775837 * 10^8$
1 século: $5{,}615692 * 10^9$

=> n³
Se n³ * 10^{-9} segundo é <= ...
1 segundo:
$\frac{n^3}{10^9} \leq 1$ //Muda o sinal de -9 para positivo
$n^3 \leq 1 * 10^9$ //Multiplica por 10^9
$\sqrt[3]{n^3} \leq \sqrt[3]{10^9}$//Raiz cúbica para os termos
$n \leq \sqrt[3]{10^9}$ //Calcule a raiz cúbica
$n \leq 10^3$
Logo, n = 10^3

1 minuto:
$\frac{n^3}{10^9} \leq 60$ //Muda o sinal de -9 para positivo
$n^3 \leq 60 * 10^9$ //Multiplica por 10^9
$\sqrt[3]{n^3} \leq \sqrt[3]{60 * 10^9}$//Raiz cúbica para os termos
$n \leq \sqrt[3]{60 * 10^9}$ //Isole 10^3 da raiz cúbica
$n \leq \sqrt[3]{60} * 10^3$ //Calcule a raiz cúbica
$n \leq 3{,}914867 * 10^3$
Logo, n = $3{,}914867 * 10^3$

1 hora:
$\frac{n^3}{10^9} \leq 3600$ //Muda o sinal de -9 para positivo

$n^3 \leq 3600 * 10^9$ //Multiplica por 10^9
$\sqrt[3]{n^3} \leq \sqrt[3]{3600 * 10^9}$//Raiz cúbica para os termos
$n \leq \sqrt[3]{3660 * 10^9}$ //Isole 10^3 da raiz cúbica
$n \leq \sqrt[3]{3600} * 10^3$ //Calcule a raiz cúbica
$n \leq 15{,}326188 * 10^3$
$n \leq 1{,}5326188 * 10^4$
Logo, n = $1{,}5326188 * 10^4$

1 dia:
$\frac{n^3}{10^9} \leq 86.400$//Muda o sinal de -9 para positivo
$n^3 \leq 86.400 * 10^9$ //Multiplica por 10^9
$\sqrt[3]{n^3} \leq \sqrt[3]{86.400 * 10^9}$//Raiz cúbica para os termos
$n \leq \sqrt[3]{86.400 * 10^9}$ //Isole 10^3 da raiz cúbica
$n \leq \sqrt[3]{86.400} * 10^3$ //Calcule a raiz cúbica
$n \leq 44{,}208377 * 10^3$
$n \leq 4{,}4208377 * 10^4$
Logo, n = $4{,}4208377 * 10^4$

1 mês:
$\frac{n^3}{10^9} \leq 2.592.000$//Muda o sinal de -9 para positivo
$n^3 \leq 2.592.000 * 10^9$ //Multiplica por 10^9
$\sqrt[3]{n^3} \leq \sqrt[3]{2.592.000 * 10^9}$//Raiz cúbica para os termos
$n \leq \sqrt[3]{2.592.000 * 10^9}$ //Organize as potências
$n \leq \sqrt[3]{2{,}592 * 10^{15}}$ //Isole 10^5 da raiz cúbica
$n \leq \sqrt[3]{2{,}592} * 10^5$ //Calcule a raiz cúbica
$n \leq 1{,}373657 * 10^5$
Logo, n = $1{,}373657 * 10^5$

1 ano:
$\frac{n^3}{10^9} \leq 31.536.000$//Muda o sinal de -9 para positivo
$n^3 \leq 31.536.000 * 10^9$ //Multiplica por 10^9
$\sqrt[3]{n^3} \leq \sqrt[3]{31.536.000 * 10^9}$//Raiz cúbica para os termos
$n \leq \sqrt[3]{31.536.000 * 10^9}$ //Organize as potências
$n \leq \sqrt[3]{31{,}536 * 10^{15}}$ //Isole 10^5 da raiz cúbica
$n \leq \sqrt[3]{31{,}536} * 10^5$ //Calcule a raiz cúbica
$n \leq 3{,}159382 * 10^5$
Logo, n = $3{,}159382 * 10^5$

1 século:
$\frac{n^3}{10^9} \leq 31.536.000.000$//Muda o sinal de -9 para positivo
$n^3 \leq 31.536.000.000 * 10^9$ //Multiplica por 10^9
$\sqrt[3]{n^3} \leq \sqrt[3]{31.536.000.000 * 10^9}$//Raiz cúbica para os termos
$n \leq \sqrt[3]{31.536.000.000 * 10^9}$ //Organize as potências
$n \leq \sqrt[3]{31{,}536 * 10^{18}}$ //Isole 10^6 da raiz cúbica
$n \leq \sqrt[3]{31{,}536} * 10^6$ //Calcule a raiz cúbica
$n \leq 3{,}159382 * 10^6$
Logo, n = $3{,}159382 * 10^6$

1 segundo: 10^3
1 minuto: $3{,}914867 * \ 10^3$
1 hora: $1{,}5326188 * \ 10^4$
1 dia: $4{,}4208377 * \ 10^4$
1 mês: $1{,}373657 * \ 10^5$
1 ano: $3{,}159382 * \ 10^5$
1 século: $3{,}159382 * 10^6$

=> 2^n
Se $2^n * 10^{-9}$ segundo é <= ...
1 segundo:
$\frac{2^n}{10^9} \leq 1$ //Muda o sinal de -9 para positivo
$2^n \leq 10^9$ // Definição $b^x = a$ equivale $x = \log_b a$
$n \leq \log_2 10^9$//Mudar a base do log para 10
$\mathrm{n} \leq \frac{\log_{10}10^9}{\log_{10}2}$$//\log_{10}2 = 0{,}3010299957$
$\mathrm{n} \leq \frac{9}{\log_{10}2}$$//\log_{10}2 = 0{,}3010299957$
$\mathrm{n} \leq \frac{9}{0{,}3010299957}$
$\mathrm{n} \leq 29{,}89735285$

Logo, n = 29,89735285
1 segundo: 29,89735285
1 minuto: $(\log_{10}60 * 10^9)/0{,}301029995 = 35{,}80424345$
1 hora: $(\log_{10}3.600 * 10^9)/0{,}301029995 = 41{,}71113405$
1 dia: $(\log_{10}86.400 * 10^9)/0{,}301029995 = 46{,}29609655$
1 mês: $(\log_{10}2.592.000 * 10^9)/0{,}301029995 = 51{,}20298714$
1 ano: $(\log_{10}31.536.000 * 10^9)/0{,}301029995 = 54{,}8078492$
1 século: $(\log_{10}31.536.000.000 * 10^9)/0{,}301029995 = 61{,}45170539$

=> 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:

```
O valor de n para o tempo 1.0: 12.0
O valor de n para o tempo 60.0: 13.0
O valor de n para o tempo 3600.0: 15.0
O valor de n para o tempo 86400.0: 16.0
O valor de n para o tempo 2592000.0: 17.0
O valor de n para o tempo 3.1536E7: 18.0
O valor de n para o tempo 3.1536E9: 20.0
```

1 segundo: 12
1 minuto: 13
1 hora: 15
1 dia: 16
1 mês: 17
1 ano: 18
1 século: 20

Tabela com os valores preenchido

Tempo t / $f(n)=10^9 ns$	**1 Segundo**	**1 Minuto * 60**	**1 Hora *3600**	**1 Dia *86400**	**1 Mês *2592000**	**1 Ano *31104000**	**1 Século *3110400000**
10^{18}	2^{10^9}	$2^{6*10^{10}}$	$2^{3,600*10^{12}}$	$2^{8,6400*10^{13}}$	$2^{2,592000*10^{15}}$	$2^{3,1536000*10^{16}}$	$2^{3,1536000000*10^{18}}$
$\sqrt{n}$	10^{18}	$3.6*10^{21}$	$1,296*10^{25}$	$7,46496*10^{27}$	$6,718464*10^{30}$	$9,94519296*10^{32}$	$9,94519296*10^{38}$
n	10^9	$6*10^{10}$	$3,6*10^{12}$	$8,64*10^{13}$	$2,592*10^{15}$	$3,1536*10^{16}$	$3,1536*10^{18}$
n lg n	$3,962000*10^7$	$1,944659*10^9$	$9,857601*10^{10}$	$2,110374*10^{12}$	$5,673115*10^{13}$	$6,411369*10^{14}$	$5,666509*10^{16}$
n^2	$3,162277*10^4$	$2,449489*10^5$	$1,897366*10^6$	$9,295160*10^6$	$5,091168*10^7$	$1,775837*10^8$	$5,615692*10^9$
n^3	10^3	$3,914867*10^3$	$1,5326188*10^4$	$4,4208377*10^4$	$1,373657*10^5$	$3,159382*10^5$	$3,159382*10^6$
2^n	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*)=$n^{2,5}$
f(*n*)=$\sqrt{2n}$
f(*n*)=$n + 10$
f(*n*)=10^n
f(*n*)=100^n
f(*n*)=$n^2 \log n$

R.:

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

1o- f(*n*)=$\sqrt{2n}$

2o- f(*n*)=$n + 10$

3o- f(*n*)=$n^2 \log n$

4o- f(*n*)=$n^{2,5}$

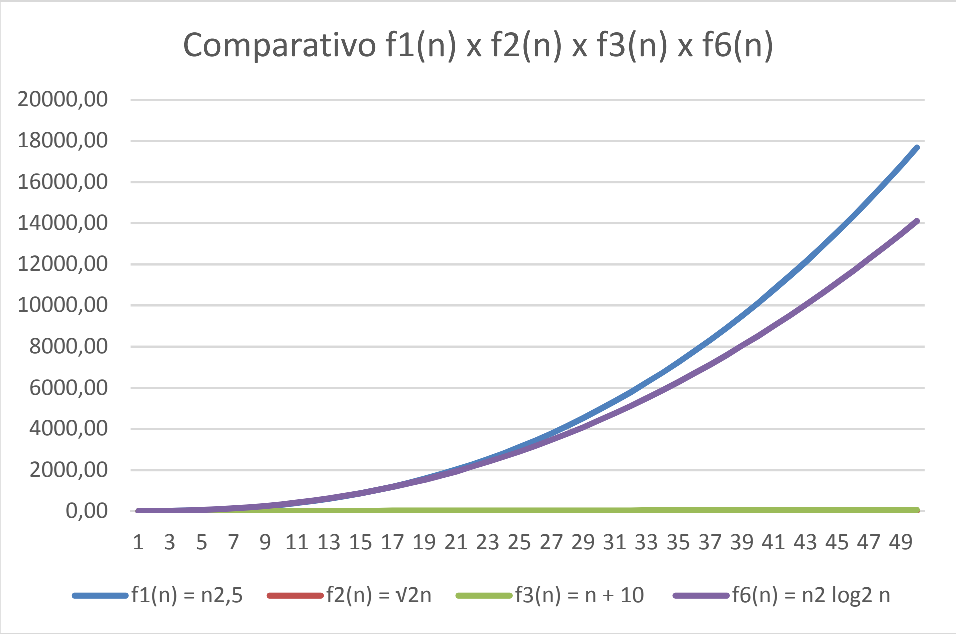
5o- f(*n*)=10^n

6o- f(*n*)=100^n

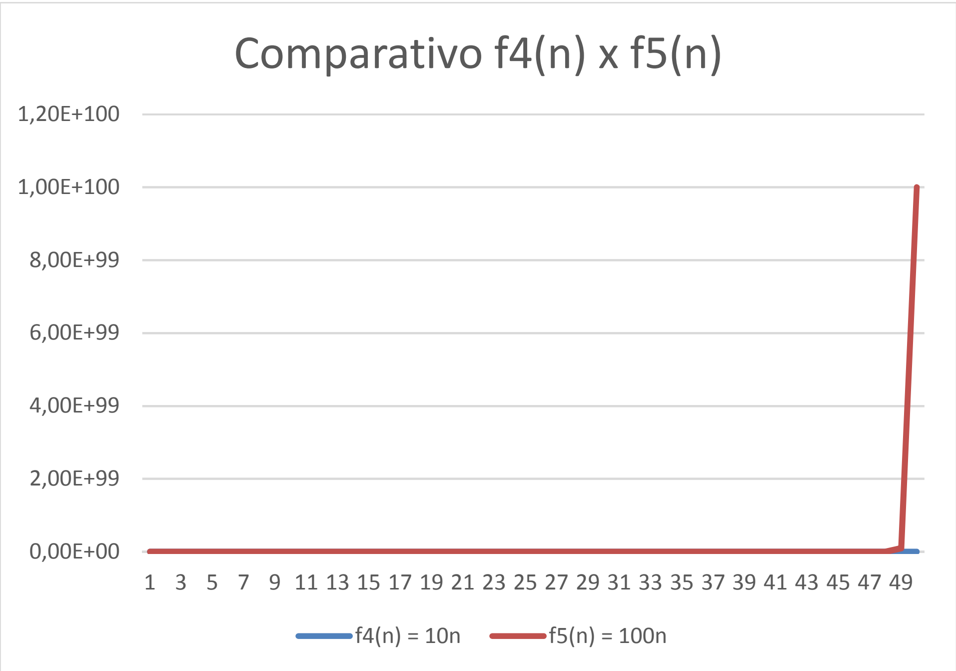
N	f1(n) = n2,5	f2(n) = √2n	f3(n) = n + 10	f4(n) = 10n	f5(n) = 100n	f6(n) = n2 log2 n
1	1,00	1,41	11,00	1,00E+01	1,00E+02	0,00
2	5,66	2,00	12,00	1,00E+02	1,00E+04	4,00
3	15,59	2,45	13,00	1,00E+03	1,00E+06	14,26
4	32,00	2,83	14,00	1,00E+04	1,00E+08	32,00
5	55,90	3,16	15,00	1,00E+05	1,00E+10	58,05
6	88,18	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)	$n - 100$	$n - 200$	$f(n) \in \Theta(g(n))$
(b)	$n^{\frac{1}{2}}$	$n^{\frac{2}{3}}$	$f(n) \in O(g(n))$
(c)	$100n + \log n$	$n + (\log n)^2$	$f(n) \in \Theta(g(n))$
(d)	$n \log n$	$10n \log 10n$	$f(n) \in \Theta(g(n))$
(e)	$\log 2n$	$\log 3n$	$f(n) \in \Theta(g(n))$
(f)	$10 \log n$	$\log(n)^2$	$f(n) \in \Theta(g(n))$
(g)	$\frac{n^2}{\log n}$	$n\ (\log n)^2$	$f(n) \in O(g(n))$
(h)	$n^{0,1}$	$(\log n)^{10}$	$f(n) \in O(g(n))$
(i)	$(\log n)^{\log n}$	$\frac{n}{\log n}$	$f(n) \in O(g(n))$
(j)	$\sqrt{n}$	$(\log n)^3$	$f(n) \in \Omega(g(n))$
(k)	$n^{\frac{1}{2}}$	$5^{\log n}$	$f(n) \in O(g(n))$
(l)	$n2^n$	3^n	$f(n) \in O(g(n))$
(m)	2^n	2^{n+1}	$f(n) \in O(g(n))$
(n)	$n!$	2^n	$f(n) \in \Omega(g(n))$
(o)	$(\log n)^{\log n}$	$2^{(\log n)^2}$	$f(n) \in \Omega(g(n))$

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

Se $0 \leq \ cg(n) \leq f(n), \exists n_o, \forall n \geq n_o$*, então f(n) ϵ Ω(g(n))*
Se $0 \leq f(n) \leq cg(n), \exists\ n_o, \forall n \geq n_o$*, então f(n) ϵ O(g(n))*
Se $f(n) \in \ O(g(n)) \land f(n) \in \Omega(g(n))$*, 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 $\frac{f(n)}{g(n)}$. Diferindo apenas o teste a ser feito, conforme a seguir:

Se $0 < \ c \leq \frac{f(n)}{g(n)}$), logo *f(n) ϵ Ω(g(n));*
Se $0 < c \land c \geq \frac{f(n)}{g(n)}$, 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 $\frac{f(n)}{g(n)}$ necessariamente deverão possuir o mesmo sinal, conforme tabela:

Produto	+	-
+	+	-
-	-	+

Note, portanto, que, se o grau de n na razão $\frac{f(n)}{g(n)}$ for:

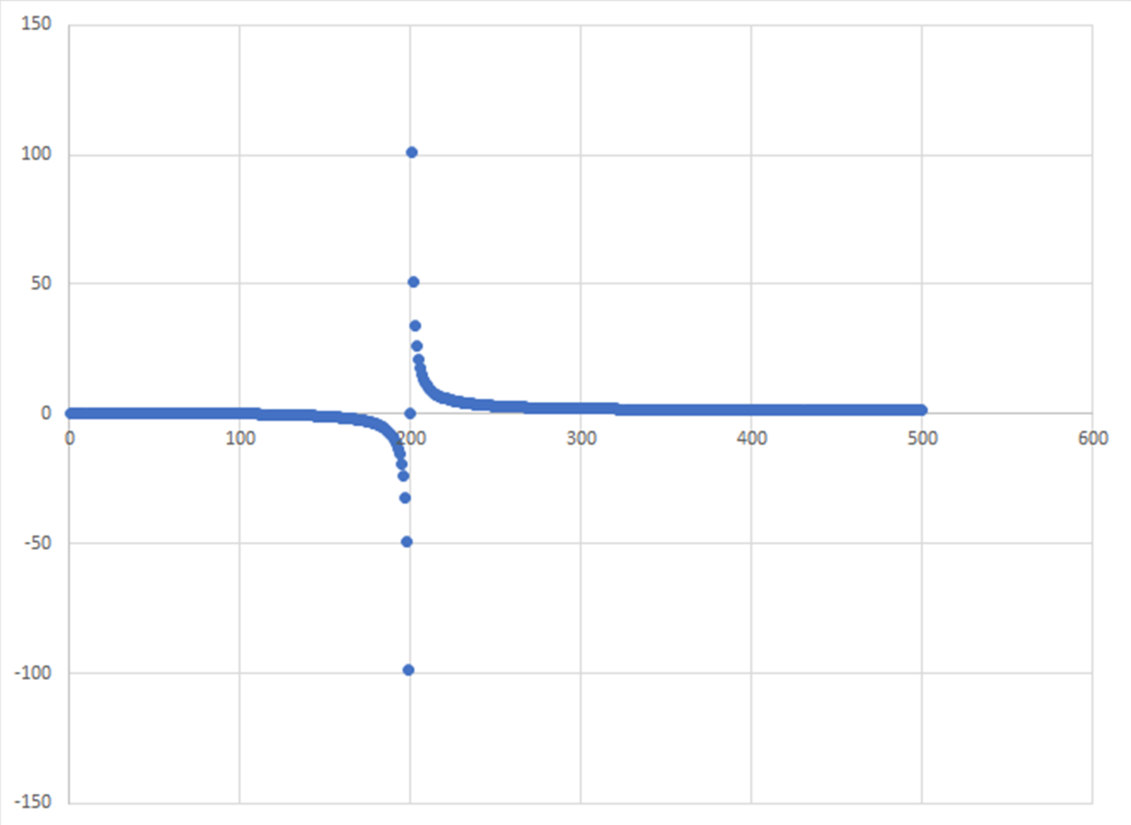
- Positivo, então *Ω(g(n));*pois $f(n)>cg(n)$, ainda que para um c positivo gigantesco;
- Negativo, então *O(g(n));*pois $f(n)<cg(n)$, ainda que para um c positivo mínimo;
- Nulo, então *Θ(g(n));*pois $f(n)>cg(n)$, ainda que para um c positivo gigantesco, E $f(n)<cg(n)$, ainda que para um c positivo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

(a) f(n)=n-100; g(n)=n-200

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{n-100}{n-200}$

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 n_0 ...

$c \leq \frac{f(n)}{g(n)}$

$c \leq \frac{n-100}{n-200}$

$2 \leq \frac{n-100}{n-200}$

$2(n-200) \leq n-100$

$2n-400 \leq n-100$

$2n-n \leq -100+400$

$n_o \leq 300$

$n_o = 300 \wedge c = 2$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq cg(n) \leq f(n)$.
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 n_0 ...

$c \geq \frac{f(n)}{g(n)}$

$c \geq \frac{n-100}{n-200}$

$2 \geq \frac{n-100}{n-200}$

$2(n-200) \geq n-100$

$2n-400 \geq n-100$

$2n-n \geq -100+400$

$n_o \geq 300$

$n_o = 300 \wedge c = 2$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$
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 c_1 e c_2 ($c_1=c_2=2$, $n_o=300$) de modo que a inequação
$0 \leq c_1(g(n)) \leq f(n) \leq c_2(g(n))$ é verdadeira

Logo ***f(n) ϵ Θ (g(n))***

(b) f(n)=$n^{\frac{1}{2}}$; g(n)=$n^{\frac{2}{3}}$

Calculando grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{n^{\frac{1}{2}}}{n^{\frac{2}{3}}}$//Fatore $n^{\frac{2}{3}}$

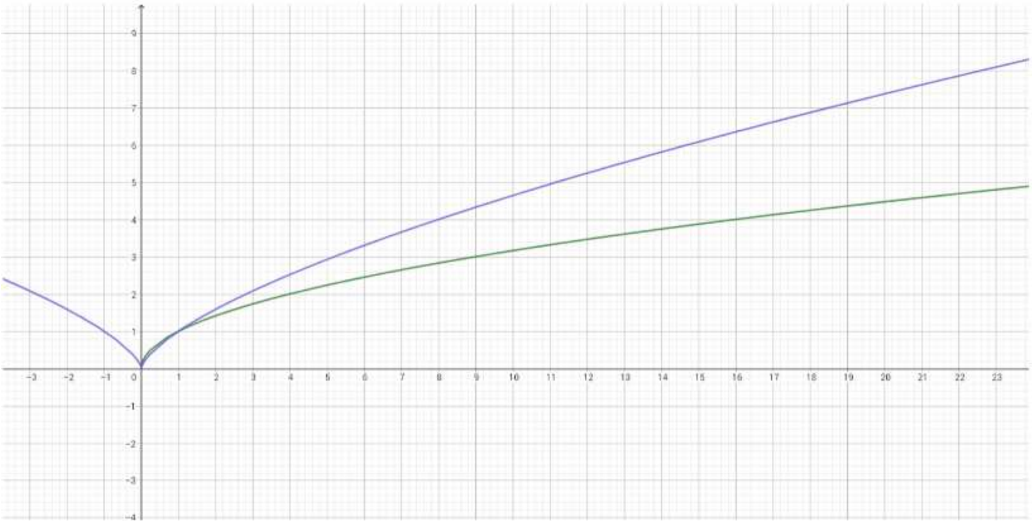
$\frac{f(n)}{g(n)} = \frac{n^{\frac{2}{3}}*n^{\frac{1}{2}*\frac{2}{3}}}{n^{\frac{2}{3}}*1}$ //MMC de 2 e 3

$\frac{f(n)}{g(n)} = \frac{n^{\frac{2}{3}}*n^{\frac{1*3-(2*2)}{6}}}{n^{\frac{2}{3}}*1}$

$\frac{f(n)}{g(n)} = \frac{n^{\frac{2}{3}}*n^{\frac{-1}{6}}}{n^{\frac{2}{3}}*1}$ //Cancele o fator comum

$\frac{f(n)}{g(n)} = n^{\frac{-1}{6}}$

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 n_0 ...

$c \leq \frac{f(n)}{g(n)}$

$c \leq= n^{\frac{-1}{6}}$

$1 \leq n^{\frac{-1}{6}}$

$1 \leq \frac{1}{n^{\frac{1}{6}}}$

$1 * n^{\frac{1}{6}} \leq 1$ // Eleve cada lado da equação a potência 6/1

$n \leq 1^{\frac{6}{1}}$ //Um elevado a qualquer potência é um

$n \leq 1$

$n_0 = 1 \wedge c = 1$

f(n) ∉ Ω (g(n)) porque não existe um c>1 e n que satisfaz a condição $0 \leq cg(n) \leq f(n)$. 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*$1^{\frac{2}{3}}$ <= $1^{\frac{1}{2}}$
0 <= 2 <= 1
0 <= 2 <= 1 = Falso

Logo, *f(n) ∉ Ω (g(n))*

f(n) ϵ O (g(n)) ?
supondo c=1, logo n_0 ...
$c \geq \frac{f(n)}{g(n)}$
$c \geq n^{\frac{-1}{6}}$
$1 \geq n^{\frac{-1}{6}}$
$1 \geq \frac{1}{n^{\frac{1}{6}}}$
$n^{\frac{1}{6}} \geq 1$ // Eleve cada lado da equação a potência 6/1
$(n^{\frac{1}{6}})^{\frac{6}{1}} \geq 1^{\frac{6}{1}}$
$n \geq 1^{\frac{6}{1}}$ //Um elevado a qualquer potência é um
$n \geq 1$
$n_0 = 1 \wedge c = 1$

c=2
$c \geq \frac{f(n)}{g(n)}$
$c \geq n^{\frac{-1}{6}}$
$2 \geq n^{\frac{-1}{6}}$ // Eleve cada lado da equação a potência -6/1
$2^{\frac{-6}{1}} \geq (n^{-\frac{1}{6}})^{-\frac{6}{1}}$
$n \leq \frac{1}{64}$
$n_0 = \frac{1}{64} \wedge c = 2$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$
Ou seja: f(n) está em O(g) para c=1 e n=1

0 <= $1^{\frac{1}{2}}$ <= 1*$1^{\frac{2}{3}}$
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))***

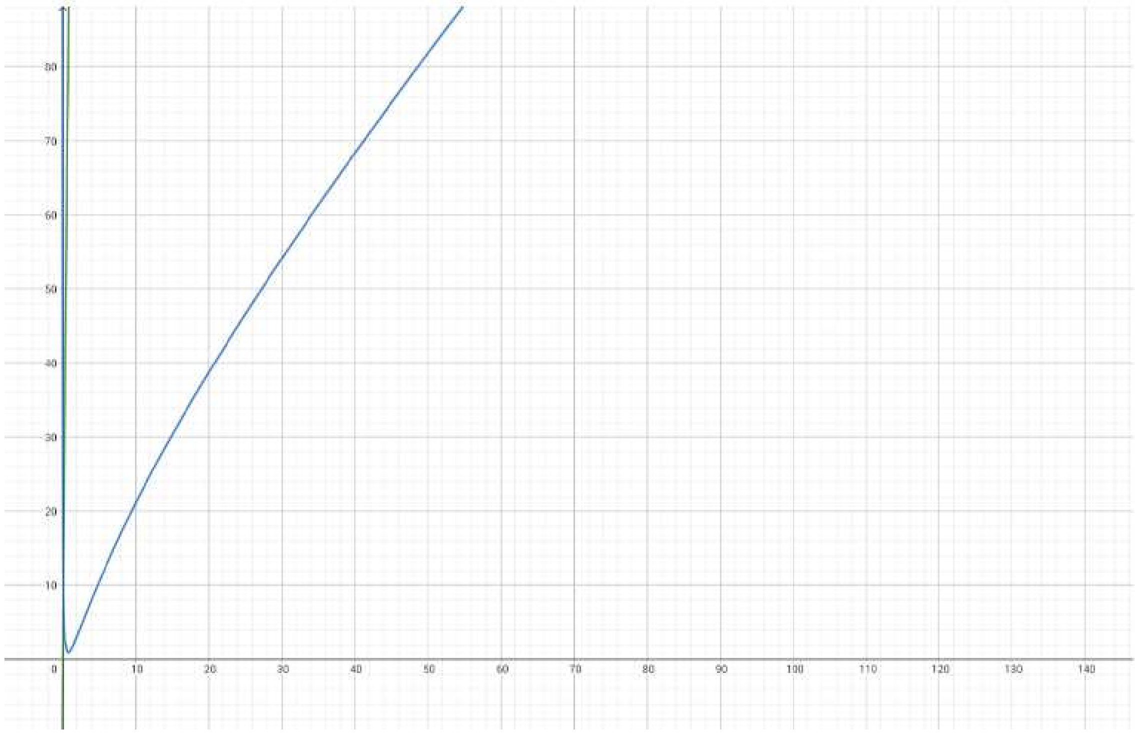
c) $\boldsymbol{f(n) = 100n + \log_2 n;\ g(n) = n + (\log_2 n)^2}$

Calculando o grau da razão $\frac{f(n)}{g(n)}$:

$\frac{f(n)}{g(n)} = \frac{100n+\log_2 n}{n+(\log_2 n)^2}$

A diferença entre os termos de maior grau na razão é 0, logo *f(n) ϵ Ω (g(n))* Λ *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 n_0=2...

$c \leq \frac{f(n)}{g(n)}$

$1 \leq \frac{100n+\log_2 n}{n+(\log_2 n)^2}$

$1 \leq \frac{100\,(2)+\log_2 2}{2+(\log_2 2)^2}$

$2 + (\log_2 2)^2 \leq 100(\,2) + \log_2 2$

$2 + 1 \leq 100(2) + 1$

$3 \leq 201$

$n_0 = 2 \wedge c = 1$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq cg(n) \leq f(n).$

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

f(n) ϵ O (g(n)) ?
supondo c=128, logo n_0=2...

$c \geq \frac{f(n)}{g(n)}$

$c \geq \frac{100n+\log_2 n}{n+(\log_2 n)^2}$

$128 \geq \frac{100(2)+\log_2 2}{2+(\log_2 2)^2}$

$128\ (2 + (\log_2 2)^2) \geq 100(2) + \log_2 2$

$128\ (2 + 1) \geq 100(2) + 1$

$128\ (3) \geq 200 + 1$

$348 \geq 201$

$n_0 = 2 \wedge c = 128$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$
Ou seja: f(n) está em O(g) para c=128 e n_o=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 c_1 e c_2 (c_1=1, c_2=128, n_o=2) de modo que a inequação
$0 \leq c_1(g(n)) \leq f(n) \leq c_2(g(n))$ é verdadeira

Logo ***f(n) ϵ Θ (g(n))***

(d) f(n)=$n\,log_2\,n$; g(n)=$10n\,log_2\,10n$

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{n\,log_2\,n}{10n\,log_2\,10n}$

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 n_0=8...

$c \leq \frac{n \log n}{10n\,log_2\,10n}$

$c \leq \frac{n\,log_2\,n}{10n + log_2 n + log_2 10}$

$0{,}01 \leq \frac{8\,log_2\,8}{10(8) * log_2 2 + log_2 10}$

$0{,}01 \leq \frac{24}{80 * 1 + 3{,}32192809}$

$0{,}01 \leq 0{,}28462347$

$n_o = 8 \wedge c = 0{,}01$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq cg(n) \leq f(n)$.
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 n_0=8...

$c \geq \frac{n \log n}{10n + log_2\,10n}$

$1 \geq \frac{n\,log_2\,n}{10n + log_2 n + log_2 10}$

$1 \geq \frac{8\,log_2\,8}{10(8) + log_2 2 + log_2 10}$

$1 \geq \frac{24}{80 + 1 + 3{,}32192809}$

$1 \geq 0{,}28462347$

$n_o = 8 \wedge c = 1$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$
Ou seja: f(n) está em O(g) para c=1 e n_o>=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 c_1 e c_2 (c_1=0,01, c_2=1, n_o=8) de modo que a inequação
$0 \leq c_1(g(n)) \leq f(n) \leq c_2(g(n))$ é verdadeira

Logo ***f(n) ϵ Θ (g(n))***

(e) f(n)=$log_2 2n$; g(n)=$log_2 3n$

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{log_2 2n}{log_2 3n}$

$\frac{f(n)}{g(n)} = \frac{log_2 2 + log_2 n}{log_2 3 + log_2 n}$

$\frac{f(n)}{g(n)} = \frac{1 + log_2 n}{1{,}58496250 + log_2 n}$

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 n_0=4...

$c \leq \frac{log_2 2n}{log_2 3n}$

$0.8 \leq \frac{log_2 2 + log_2 n}{log_2 3 + log_2 n}$

$0.8 \leq \frac{1 + log_2 n}{log_2 3 + log_2 n}$

$\mathrm{n_0} \leq 3$

$n_o = 3 \wedge c = 0.8$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq cg(n) \leq f(n).$

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 n_0=4...

$c \geq \frac{log_2 2n}{log_2 3n}$

$0.8 \geq \frac{log_2 2 + log_2 n}{log_2 3 + log_2 n}$

$0.8 \geq \frac{1 + log_2 n}{log_2 3 + log_2 n}$

$n_0 = 4 \wedge c = 1$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$

Ou seja: f(n) está em O(g) para c=1 e n_o=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)=$10\ log_2\ n$; g(n)=$log_2(n)^2$

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{10\ log_2 n}{log_2(n)^2}$

$\frac{f(n)}{g(n)} = \frac{10\ log_2 n}{2log_2(n)}$

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

A diferença entre os termos de maior grau na razão é 0, logo *f(n) ϵ Ω (g(n))* Λ *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 n_0=16 ....

$c \leq \frac{10\ log_2 n}{log_2(n)^2}$

$1 \leq \frac{10\ log_2 n}{log_2(16)^2}$

$1 \leq\ 5$

$n_0 = 16\ \Lambda\ c = 1$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq\ cg(n) \leq f(n).$

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

f(n) ϵ O (g(n)) ?

supondo c=10, logo n_0=16...

$c \geq \frac{10\ log_2 n}{log_2(n)^2}$

$10 \geq \frac{10\ log_2 n}{log_2(n)^2}$

$10 \geq \frac{10\ log_2 16}{2log_2(n)}$

$10 \geq\ 5$

$n_o = 16\ \Lambda\ c = 10$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$

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 c_1 e c_2 (c_1=1, c_2=10, n_o=16) de modo que a inequação

$0 \leq c_1(g(n)) \leq f(n) \leq c_2(g(n))$ é verdadeira

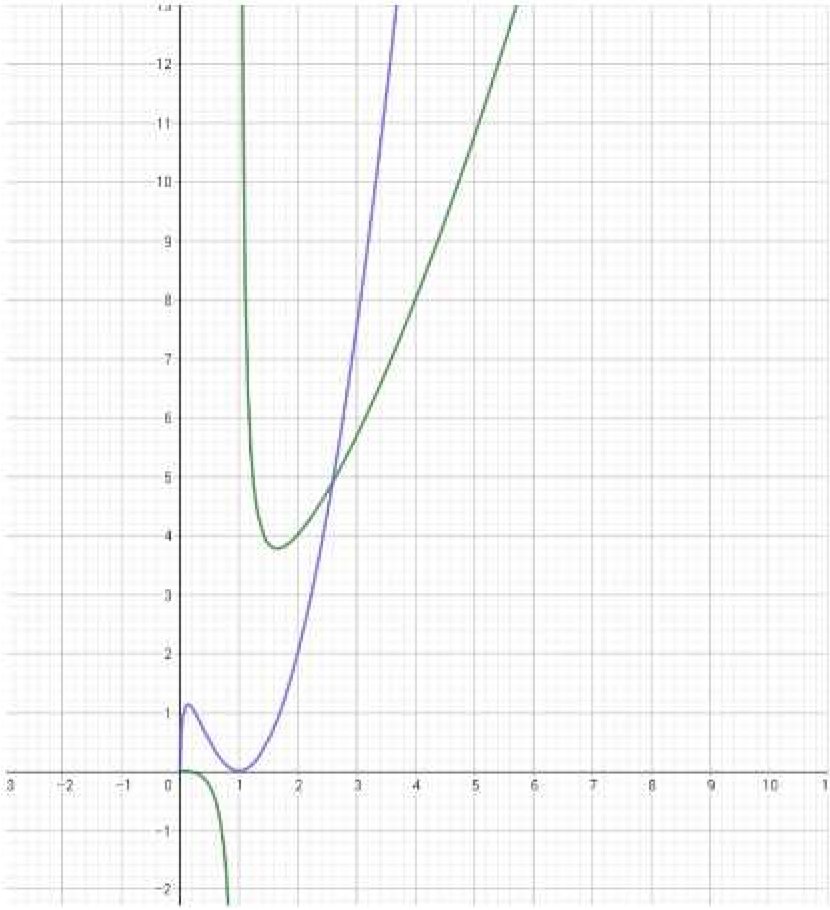
Logo ***f(n) ϵ Θ (g(n))***

(g) $f(n)=\frac{n^2}{log_2 n}$; $g(n)=n\ (log_2 n)^2$ *f(n)* ϵ Ω(g(n))

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$$\frac{f(n)}{g(n)} = \frac{\frac{n^2}{log_2 n}}{\mathrm{n}\ (log_2 n)^2}$$
$$\frac{f(n)}{g(n)} = \frac{n^2}{log_2 n} * \frac{1}{\mathrm{n}\ (log_2 n)^2}$$
$$\frac{f(n)}{g(n)} = \frac{n}{(log_2 n)^3}$$

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 n_0 =16, ...
$$c \geq \frac{n}{(log_2 n)^3}$$
$$0.3 \geq \frac{16}{(log_2 16)^3}$$

$0.3 \geq \frac{16}{4^3}$
$0.3 \geq \frac{1}{4}$
$0.3 \geq 0.25$

$n_0 = 16 \wedge c = 0.3$
f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$
Ou seja: f(n) está em O(g) para c=0.3 e n_o=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))***

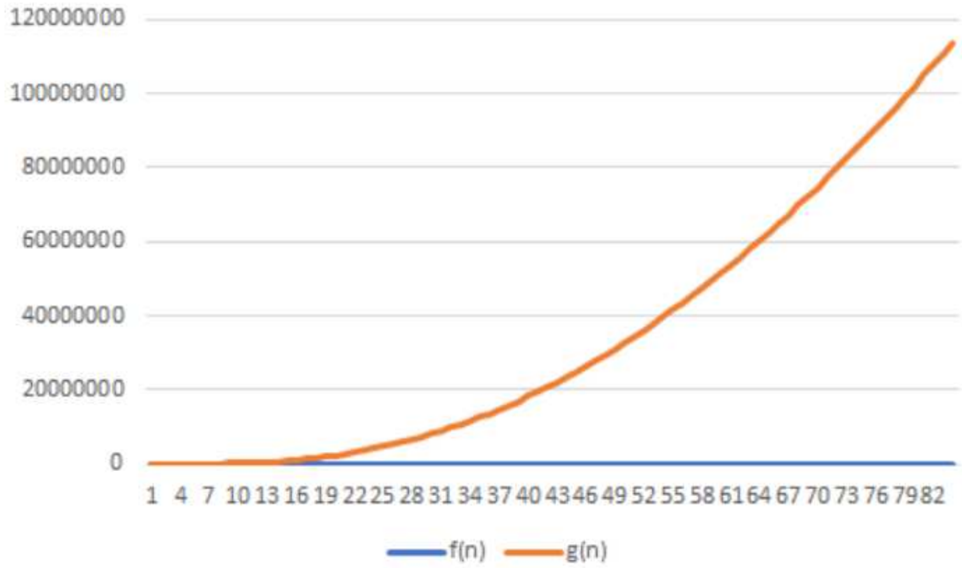
Logo, ***f(n) ϵ O (g(n))***

(h) f(n)=$n^{0,1}$; g(n)=$(log_2 n)^{10}$

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{n^{0,1}}{(log_2 n)^{10}}$

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 n_0 =4, ...

$c \geq \frac{n^{0,1}}{(log_2 n)^{10}}$

$c \geq \frac{4^{0,1}}{(log_2 4)^{10}}$

$c \geq \frac{\sqrt[10]{4}}{2^{10}}$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$
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))***

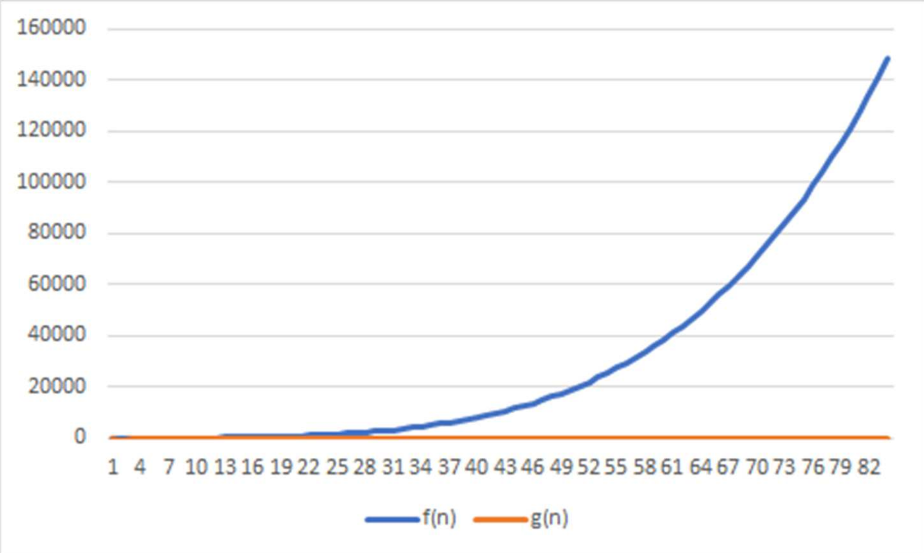
Logo, ***f(n) ϵ O (g(n))***

(i) f(n)=$(log_2 n)^{log_2 n}$; g(n)=$\frac{n}{log_2 n}$

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{(log_2 n)^{log_2 n}}{\frac{n}{log_2 n}}$

Comparando graficamente as funções f(n)=$(log_2 n)^{log_2 n}$ e g(n)=$\frac{n}{log_2 n}$ 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 n_0 =2

$c \geq \frac{(log_2 n)^{log_2 n}}{\frac{n}{log_2 n}}$

$c \geq \frac{(log_2 2)^{log_2 2}}{\frac{2}{log_2 2}}$

$c \geq \frac{1}{\frac{2}{1}}$

$c \geq \frac{1}{2}$

$n_0 = 2 \wedge c = 1$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$

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))***

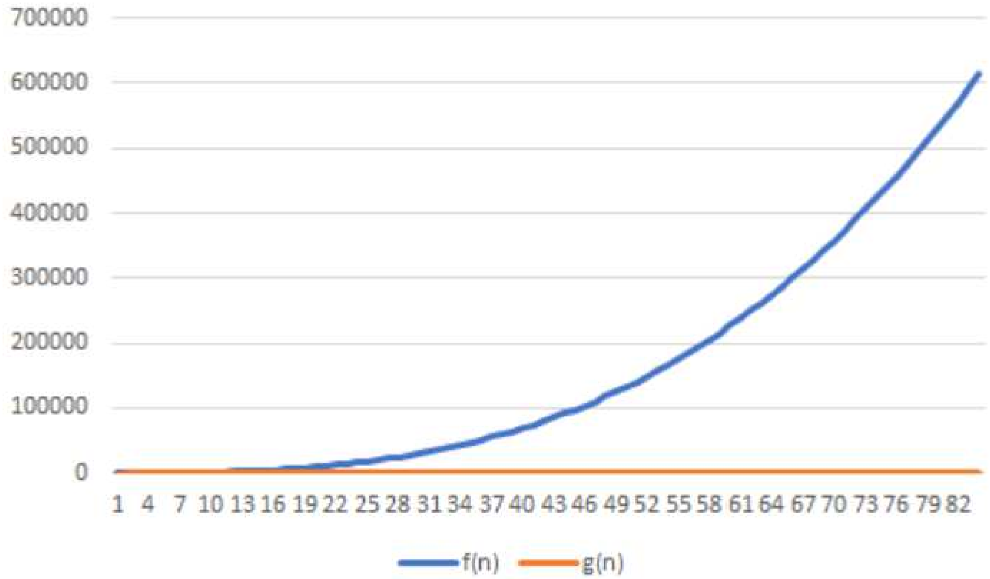
Logo, ***f(n) ϵ O (g(n))***

(j) f(n)=$\sqrt{n}$; g(n)=$(log_2 n)^3$

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{\sqrt{n}}{(log_2 n)^3}$

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 n_0 =4...

$c \leq \frac{\sqrt{n}}{(log_2 n)^3}$

$0.1 \leq \frac{\sqrt{4}}{(log_2 4)^3}$

$0.1 \leq \frac{2}{2^3}$

$0.1 \leq 0.25$

$n_0 = 4 \wedge c = 0.1$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq cg(n) \leq f(n)$.

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))***

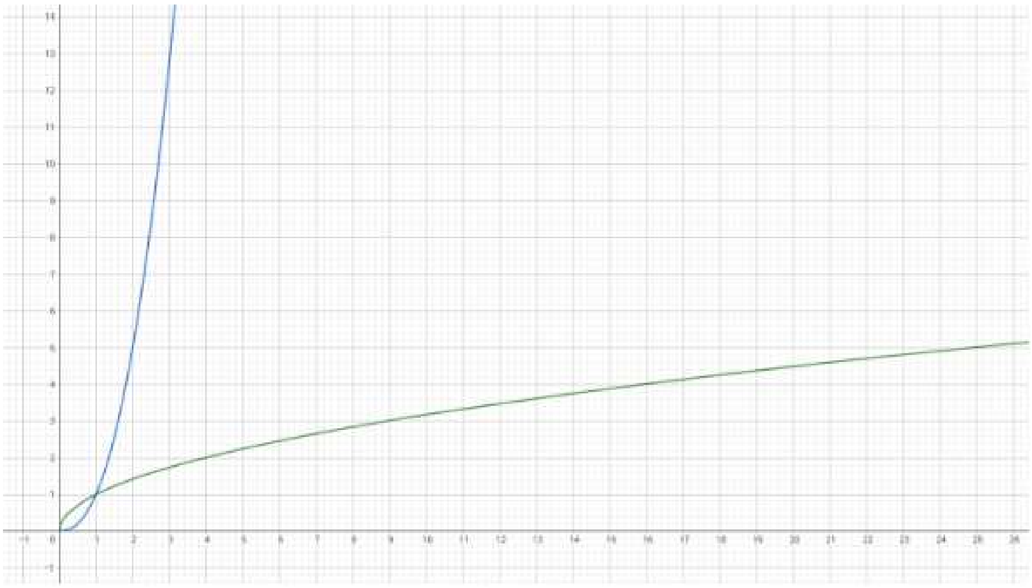
Logo, ***f(n) ϵ*** Ω ***(g(n))***

(k) f(n)=$n^{\frac{1}{2}}$; g(n)=$5^{\log n}$

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$$\frac{f(n)}{g(n)} = \frac{n^{\frac{1}{2}}}{5^{log_2 n}}$$

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 n_0=16...

$c \geq \frac{\sqrt{n}}{5^{log_2 n}}$

$0.05 \geq \frac{\sqrt{16}}{5^{log_2 16}}$

$0.05 \geq \frac{4}{5^4}$

$0.05 \geq 0{,}0064$

$n_0 = 16 \wedge c = 0.05$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$

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))***

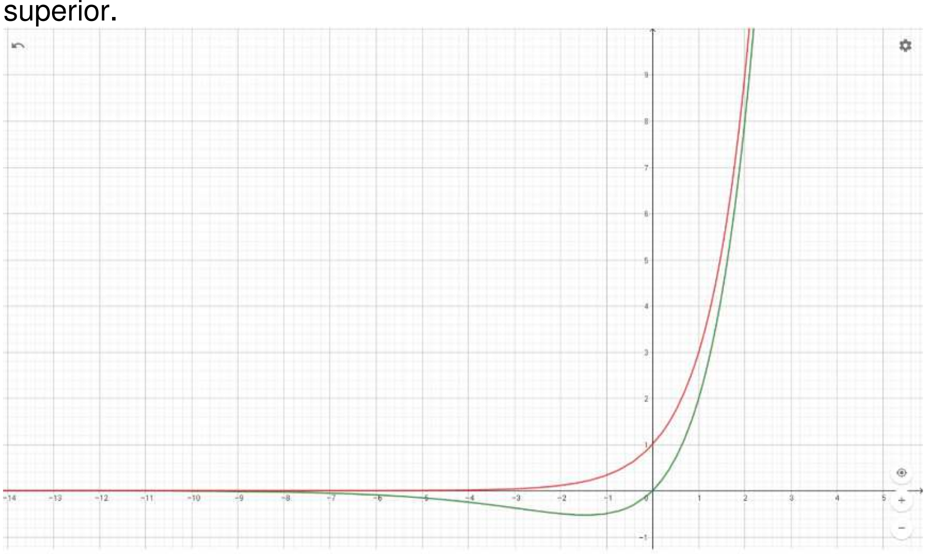
Logo, ***f(n) ϵ** O **(g(n))***

(l) f(n)=$n2^n$; g(n)=3^n

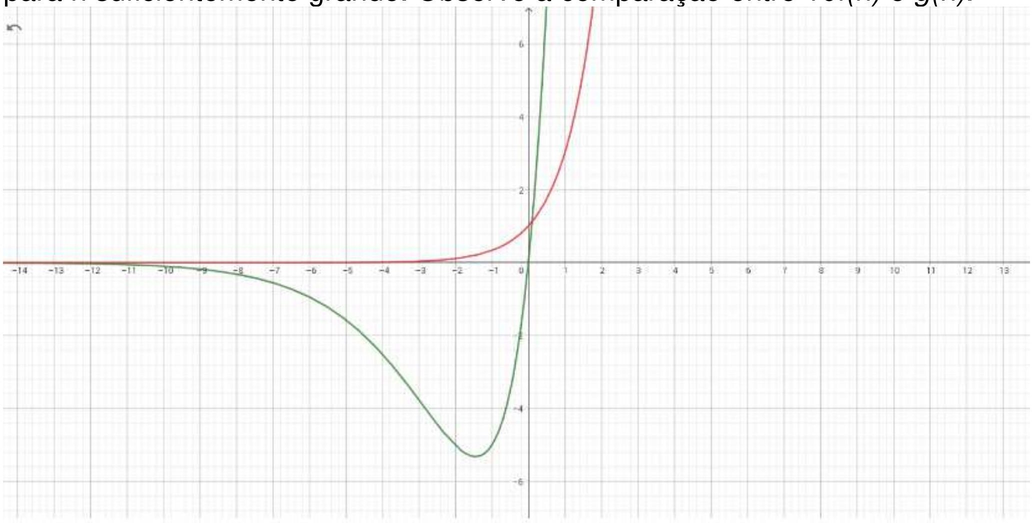
Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{n2^n}{3^n}$

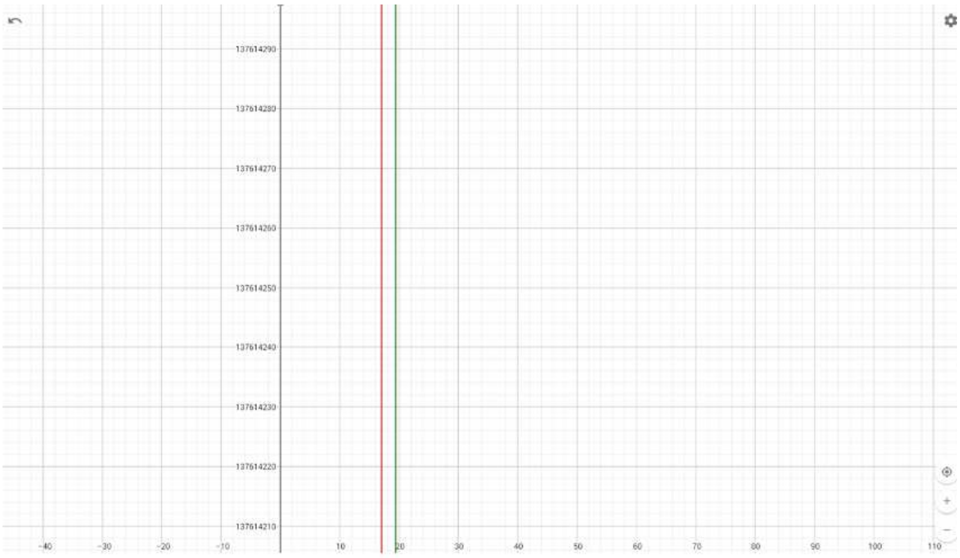
Comparando graficamente as funções f(n) e g(n) identificamos que g(n) possui um comportamento assintótico muito superior ao de f(n), de forma que a ordem de g(n) é superior.



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 3^n como um binômio a fim de melhor estimar o crescimento para n suficientemente grande em relação a 2^n, ou seja, $(2+1)^n$.

$$\frac{n2^n}{(1+2)^n}$$

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

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1
1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1
1 15 105 455 1365 3003 5005 6435 6435 5005 3003 1365 455 105 15 1
1 16 120 560 1820 4368 8008 11440 12870 11440 8008 4368 1820 560 120 16 1
1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1
1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1
1 19 171 969 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 969 171 19 1
1 20 190 1140 4845 15504 38760 77520 125970 167960 184756 167960 125970 77520 38760 15504 4845 1140 190 20 1
```

Existe uma regra de formação onde os coeficientes da linha L determinam os coeficientes da linha L+1.
Exemplo:
$(x+y)^2 = x^2 + 2xy + y^2 = 1x^2y^0 + 2x^1y^1 + 1x^0y^2$

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

Quando um binômio é da forma $(x+y)$, elevado a uma potência, temos:
$(x+y)^n = a_0x^n + a_1x^{n-1}y + a_2x^{n-2}y^2 + \cdots + a_{n-1}xy^{n-1} + a_ny^n$

onde a_i 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)*** $\notin$ Ω ***(g(n))***

f(n) ϵ O (g(n)) ?
supondo c=0.8 logo n_0=4...
$c \geq \frac{f(n)}{g(n)}$
$c \geq \frac{n2^n}{3^n}$
$0.8 \geq \frac{4*2^4}{3^4}$
$0.8 \geq \frac{64}{81}$
$0.8 \geq 0{,}790123$
$n_0 = 4 \wedge c = 0.8$
f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$
Ou seja: f(n) está em O(g) para c=0.8 e n=14

f(n) ϵ Θ (g(n)) ?
Como *f(n)* $\notin$ Ω *(g(n))* então ***f(n)*** $\notin$ ***Θ (g(n))***

Logo, ***f(n) ϵ** O **(g(n))***

(m) f(n)=2^n; g(n)=2^{n+1} *f*(*n*) ϵ Θ(g(n))

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{2^n}{2^{n+1}}$

$\frac{f(n)}{g(n)} = \frac{2^n}{2^n * 2^1}$

$\frac{f(n)}{g(n)} = \frac{2^n}{2^n * 2^1}$//Cancela 2^n

$\frac{f(n)}{g(n)} = \frac{1}{2}$

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

***f*(*n*) ϵ Ω(g(n)) ?**

supondo c=0.1, logo n_0=4...

$c \leq \frac{2^n}{2^{n+1}}$

$0.1 \leq \frac{2^4}{2^{4+1}}$

$0.1 \leq \frac{16}{32}$

$\mathrm{n}_0 \leq 0{,}5$

$n_o = 4 \wedge c = 0.1$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq cg(n) \leq f(n)$.
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 n_0=4...

$c \geq \frac{2^n}{2^{n+1}}$

$1 \geq \frac{2^4}{2^{4+1}}$

$1 \geq \frac{16}{32}$

$1 \geq 0{,}5$

$n_0 = 4 \wedge c = 1$

f(n) ϵ *O* (g(n)) porque existe um c e n que satisfaz a condição $0 \leq f(n) \leq cg(n)$
Ou seja: f(n) está em O(g) para c=1 e n_o=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 c_1 e c_2 (c_1=0.1, c_2=1, n_o=4) de modo que a inequação
$0 \leq c_1(g(n)) \leq f(n) \leq c_2(g(n))$ é verdadeira

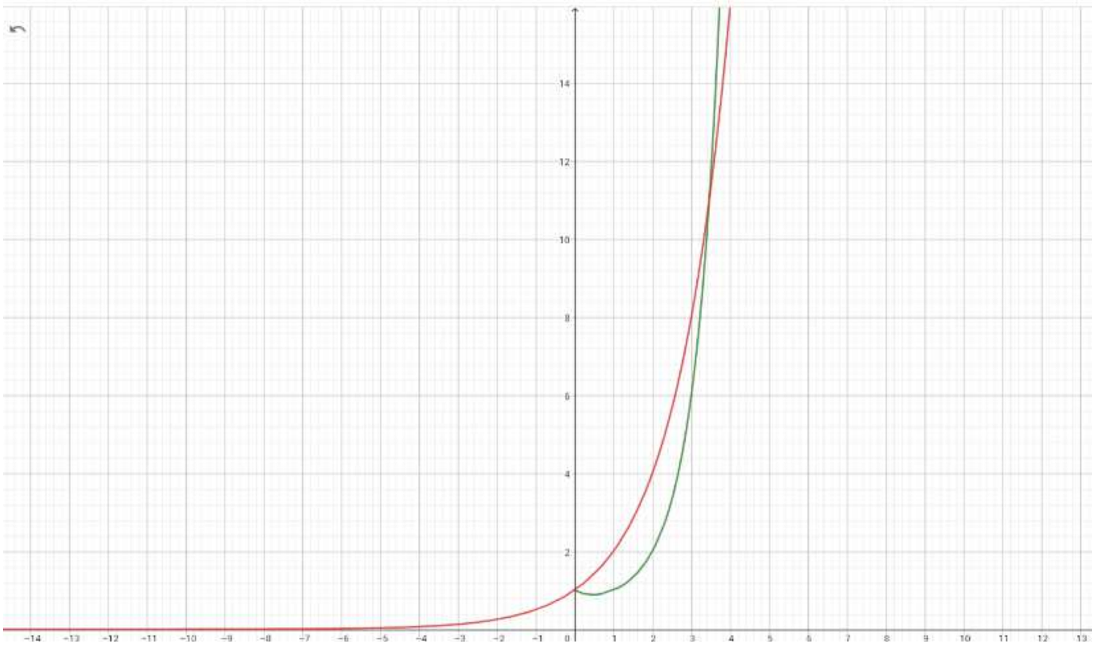
Logo ***f(n) ϵ Θ (g(n))***

(n) f(n)=$n!$; g(n)=2^n *f(n)* ϵ Ω (g(n))

Calculando o grau da razão $\frac{f(n)}{g(n)}$

$\frac{f(n)}{g(n)} = \frac{n!}{2^n}$

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 n_0 =4...

$c \leq \frac{n!}{2^n}$

$1 \leq \frac{4!}{2^4}$

$1 \leq \frac{4*3*2*1}{16}$

$1 \leq \frac{24}{16}$

$1 \leq 1.5$

$n_0 = 4 \wedge c = 1$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq cg(n) \leq f(n)$.
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)=$(\boldsymbol{log_2 n})^{\boldsymbol{log_2 n}}$; g(n)=$2^{(\boldsymbol{log_2 n})^2}$

Calculando o grau da razão $\frac{f(n)}{g(n)}$

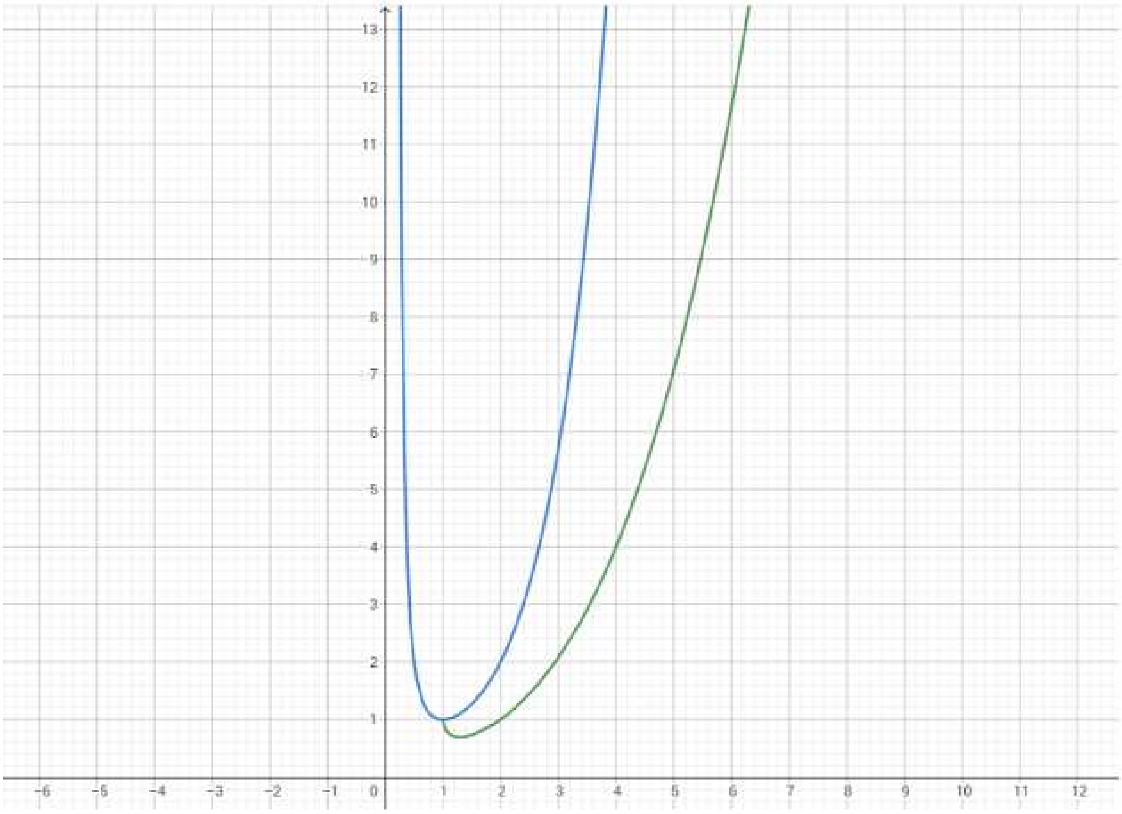
$\frac{f(n)}{g(n)} = \frac{(\boldsymbol{log_2 n})^{\boldsymbol{log_2 n}}}{2^{(\boldsymbol{log_2 n})^2}}$

Suponha $\boldsymbol{log_2 n = x}$

$\frac{f(n)}{g(n)} = \frac{\boldsymbol{x^x}}{\boldsymbol{2^{x^2}}}$

$\frac{f(n)}{g(n)} = \frac{(\boldsymbol{2+(x-2)})^{\boldsymbol{x}}}{\boldsymbol{2^x * 2^x}}$

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 n_0=8...

$c \leq \frac{(\boldsymbol{log_2 n})^{\boldsymbol{log_2 n}}}{\boldsymbol{2^{(log_2 n)^2}}}$

$1 \leq \frac{(\boldsymbol{log_2 8})^{\boldsymbol{log_2 8}}}{\boldsymbol{2^{(log_2 8)^2}}}$

$1 \leq \frac{\boldsymbol{3^3}}{\boldsymbol{2^{3^2}}}$

$1 \leq \frac{\boldsymbol{3^3}}{\boldsymbol{2^{3^2}}}$

$1 \leq 0{,}05273437$
$n_o = 8 \wedge c = 1$

f(n) ϵ Ω (g(n)) porque existe um c e n que satisfaz a condição $0 \leq\ cg(n) \leq f(n).$
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))***