

Lista de Exercícios

- Ordene as seguintes funções por ordem de crescimento. Coloque a relação de O ou Θ entre cada par da sequência ordenada (considere as bases dos logaritmos como sendo 2).
 $\log(\log n)$, $2^{\log n}$, $\log_3(n)$, $(\sqrt{2})^{\log n}$, n^2 , $n!$, $(\log n)!$, $(\frac{3}{2})^n$, n^3 , $(\log n)^2$, $(\log n!)$, $\log(n!)$, 2^{2^n} , $n^{1/\log n}$, $n \cdot 2^n$, $n^{\log \log n}$, $\log n$, 5, $2^{\log n}$, $(\log n)^{\log n}$, e^n , $4^{\log n}$, $(n+1)!$, $\sqrt{\log n}$, $2^{\sqrt{2} \log n}$, n , $2n$, $n \log n$, 2^{2n+1} , 2^{n-1} , 2^{3n} , 2^5 , $2^{\frac{n}{2}}$
 OBS: não é necessário entregar as demonstrações, mas faça a demonstração da relação entre cada par de funções para praticar demonstrações!
- Sejam $f(n)$ e $g(n)$ funções assintoticamente não negativas. Usando a definição básica da notação Θ , prove que $\max(f(n), g(n)) = \Theta(f(n) + g(n))$
- Analise (com detalhes) a complexidade de pior caso dos algoritmos abaixo.

ALG1

Entrada Um tamanho de problema n .

```

for  $i := 1 \dots n$  do
  for  $j := i \dots 2^i$ 
    operações com custo  $O(i)$ 
  end for
end for

```

ALG2

Entrada Um tamanho de problema n .

```

 $i := 1$ 
while  $i \leq n$  do
   $j := 1$ 
  while  $j \leq 2^i$ 
    operações com custo  $O(j)$ 
     $j := j * 2$ 
  end for
   $i := i + 1$ 
end for

```