Lista de Exercícios

1. Ordene as seguintes funções por ordem de crescimento. Coloque a relação de O ou Θ entre cada par da seqüencia ordenada (considere as bases dos logarítmos 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!

   Observe que:

$$
\begin{aligned}
n^{\frac{1}{\log_2 n}} &== n^{\frac{\log_2 2}{\log_2 n}} = n^{\log_n 2} = 2^{\log_n n} = 2\\
2^{\sqrt{2\log n}} &= 2^{\sqrt{2}\cdot\sqrt{\log n}}\\
\sqrt{2}^{\log_2 n} &= n^{\log_2 \sqrt{2}} = n^{\frac{1}{2}} = \sqrt{n}\\
2^{\log_2 n} &= n\\
4^{\log n} &= n^{\log 4} = n^2\\
2^{\frac{n}{2}} &= (2^{\frac{1}{2}})^n = (\sqrt{2})^n \approx (1.41)^n\\
e^n &\approx (2.72)^n\\
2^{n-1} &= \frac{2^n}{2}\\
2^{2n+1} &= 2\cdot 4^n\\
2^{3n} &= 8^n\\
n^{\log(\log n)} &= (\log n)^{(\log n)} \qquad \text{propriedade dos expoentes dos logarítmos}\\
\log(n!) &\approx n\log n - \frac{n}{\log_e 2} + \frac{1}{2}\log n + O(1) \qquad \text{livro da Laira, } 2^a \text{ edição, pág.20}\\
n! &\approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n (1+\Theta(1/n)) \qquad \text{Cormen tradução } 2^a \text{ edição americana, pág 44}\\
(\log n)! &\approx \sqrt{2\pi\log n}\left(\frac{\log n}{e}\right)^{\log n}(1+\Theta(1/\log n)) \qquad \text{substitui n por } \log n \text{ na fórmula anterior}\\
(n+1)! &= (n+1)\cdot n!
\end{aligned}
$$

   Ordem:

   1) $5 \in \Theta(2^5)$
   2) $2^5 \in \Theta(n^{\frac{1}{\log n}})$
   3) $n^{\frac{1}{\log n}} \in O(\log(\log n))$
   4) $(\log(\log n)) \in O(\sqrt{\log n})$
   5) $(\sqrt{\log n}) \in O(\log_3 n)$
   6) $(\log_3 n) \in \Theta(\log n)$
   7) $(\log n) \in O(2^{\sqrt{2\log n}})$
   8) $((2^{\sqrt{2\log n}}) \in O(\sqrt{n})$
   9) $(\sqrt{n}) \in \Theta(\sqrt{2}^{\log n})$
   10) $\sqrt{2}^{\log n} \in O(\log n)^2$

11) $(\log n)^2 \in O(2^{\log n})$
12) $(2^{\log n}) \in \Theta(n)$
13) $n \in \Theta(2n)$
14) $2n \in O(\log(n!))$
15) $(\log n!) \in \Theta(n \log n)$
16) $(n \log n) \in O(4^{\log n})$
17) $4^{\log n} \in \Theta(n^2)$
18) $n^2 \in O(n^3)$
19) $n^3 \in O((\log n)!)$
20) $(\log n)! \in O(n^{\log \log n})$
21) $(n^{\log \log n}) \in \Theta(\log n)^{\log n}$
22) $(\log n)^{(\log n)} \in O(2^{\frac{n}{2}})$
23) $2^{\frac{n}{2}} \in O(\frac{3}{2}^n)$
24) $(\frac{3}{2})^n \in O(2^{n-1})$
25) $2^{n-1} \in O(n \cdot 2^n)$
26) $(n \cdot 2^n) \in O(e^n)$
27) $O(e^n) \in O(2^{2n+1})$
28) $(2^{2n+1}) \in O(2^{3n})$
29) $(2^{3n}) \in O(n!)$
30) $(n!) \in O(n+1)!$
31) $(n+1)! \in O(2^{2^n})$

2. Sejam $f(n)$ e $g(n)$ funções assintoticamente não negativas. Usando a definição básica da notação Θ, prove que $\max(f(n), g(n)) = \Theta(f(n) + g(n))$

Sabe-se que $f(n) + g(n) = f(n) + g(n)$ Então para funções não negativas:

$g(n) \leq f(n) + g(n)$

$f(n) \leq f(n) + g(n)$

$max(f(n), g(n)) \leq c(f(n) + g(n))$ para c=1 e $\forall n_0 \geq 0$. Então:

$max(f(n), g(n)) \in O(f(n) + g(n))$

Sabe-se que $f(n) + g(n) = f(n) + g(n)$ Então para funções não negativas:

$max(f(n), g(n)) \geq c(f(n) + g(n))$ para c=$\frac{1}{2}$ e $\forall n_0 \geq 0$. Então:

$max(f(n), g(n)) \in \Omega(f(n) + g(n))$

Como $max(f(n), g(n)) \in O(f(n)+g(n)$ e $max(f(n), g(n)) \in \Omega(f(n)+g(n)$, então $max(f(n), g(n)) \in \Theta(f(n) + g(n)$.

3. Analise (com detalhes) a complexidade de pior caso dos algoritmos abaixo.

Algoritmo 0.1 (Alg1)
Entrada Um tamanho de problema n.

```
for i := 1...n do
  for j := i...2^i
    operações com custo O(i)
  end for
end for
```

then

$$
\begin{aligned}
C_p[ALG1](n) &= \sum_{1\le i\le n} \sum_{i\le j\le 2^i} i \\
&= \sum_{1\le i\le n} (2^i - i + 1)i \\
&= \sum_{1\le i\le n} i\cdot 2^i - \sum_{1\le i\le n} i^2 + \sum_{1\le i\le n} i \\
&= 2 + (n-1)\cdot 2^{n+1} - \frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \\
&= \frac{12 + 12n2^n - 12\cdot 2^n - 2n^3 - 3n^2 - n + 3n^2 + 3n}{6}
\end{aligned}
$$

Como o termo de mais altar ordem é $\frac{12n2^n}{6}$, a complexidade do algoritmo é $\Theta(n2^n)$

Algoritmo 0.2 (Alg2)
Entrada Um tamanho de problema n.

```
i := 1
while i ≤ n do
  j := 1
  while j ⇐ 2^i
    operações com custo O(j)
    j := j * 2
  end while
  i := i + 1
end while
```

$$
\begin{aligned}
C_p[ALG2](n) &= C_p[l_1] + C_p[l_2 ... l_9] \\
C_p[l_2 ... l_9] &= C_p[l_3] + C_p[l_4 ... l_7] + C_p[l_8] \\
C_p[l_4 - l_7] &= C_p[l_5] + C_p[l_6] \\
\text{then} & \\
C_p[ALG2](n) &= 1 + \sum_{1 \le i \le n} (1 + \sum_{0 \le k \le \log_2 2^i} (2^k + 1) + 1) \\
&= 1 + \sum_{1 \le i \le n} (2 + \sum_{0 \le k \le i} (2^k + 1)) \\
&= 1 + \sum_{1 \le i \le n} (2 + 2^{i+1} - 1 + i + 1) \\
&= 1 + \sum_{1 \le i \le n} (2 + 2^{i+1} + i) \\
&= 1 + \sum_{1 \le i \le n} 2 + \sum_{1 \le i \le n} 2 \cdot 2^i + \sum_{1 \le i \le n} i \\
&= 1 + 2n + 2 \cdot \sum_{1 \le i \le n} 2^i + \sum_{1 \le i \le n} i \\
&= 1 + 2n + 2(2^{n+1} - 2) + \frac{n(n+1)}{2} \\
&= 4 \cdot 2^n - 3 + n^2/2 + 5n/2
\end{aligned}
$$

Como o termo de mais altar ordem é $4 \cdot 2^n$, a complexidade do algoritmo é $\Theta(2^n)$