

# Máximo

$$\begin{aligned} \text{maximum}[2, 5, 1] = \\ \max 2 \left( \begin{aligned} \text{maximum}[5, 1] = \\ \max 5 \left( \begin{aligned} \text{maximum}[1] = \\ 1 \end{aligned} \right) \end{aligned} \right) \end{aligned}$$

Fonte: <http://haskell.tailorfontela.com.br/recursion>

PF 2.2 e 2.3

<http://www.ime.usp.br/~pf/algoritmos/aulas/recu.html>

# Problema do máximo

**Problema:** encontrar o valor de um elemento máximo de um vetor  $v[0:n-1]$ .

Entra:

	0			4						$n-1$	
v	10	44	10	35	99	40	20	65	55	50	38

Sai: máximo == 99

## Máximo recursivo

```
def maximoR(n,v):  
    '''(int,list) -> item  
    Recebe um inteiro n e uma lista v,  
    e retorna o maio elemento das n primeiras  
    posições de v.  
    '''  
    if n == 1:  
        return v[0]  
    else:  
        x = maximoR(n-1, v)  
        if x > v[n-1]:  
            return x  
        else  
            return v[n-1]
```

...alternativamente ...

```
def maximoR(n,v):  
    '''(int,list) -> item  
    Recebe um inteiro n e uma lista v,  
    e retorna o maior elemento das n  
    primeiras posições de v  
    '''  
  
    if n == 1:  
        return v[0]  
    x = maximoR(n-1, v)  
    if x > v[n-1]:  
        return x  
    return v[n-1]
```

## Outro máximo recursivo

```
def maximo(n, v):  
    return maximoR(0, n, v)  
  
def maximoR(i, n, v):  
    if i == n-1:  
        return v[i]  
    else:  
        x = maximoR(i+1, n, v)  
        if x > v[i]:  
            return x  
        else:  
            return v[i]
```

... alternativamente ...

```
def maximo(n, v):  
    return maximoR(0, n, v)
```

```
def maximoR(i, n, v):  
    if i == n-1:  
        return v[i]  
    x = maximoR(i+1, n, v)  
    if x > v[i]:  
        return x  
    return v[i]
```