

RECURSÃO. SOLUÇÕES AOS EXERCÍCIOS

Santiago Valdés Ravelo
[https://ic.unicamp.br/~santiago/
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

MC102 - Algoritmos e
Programação de
Computadores

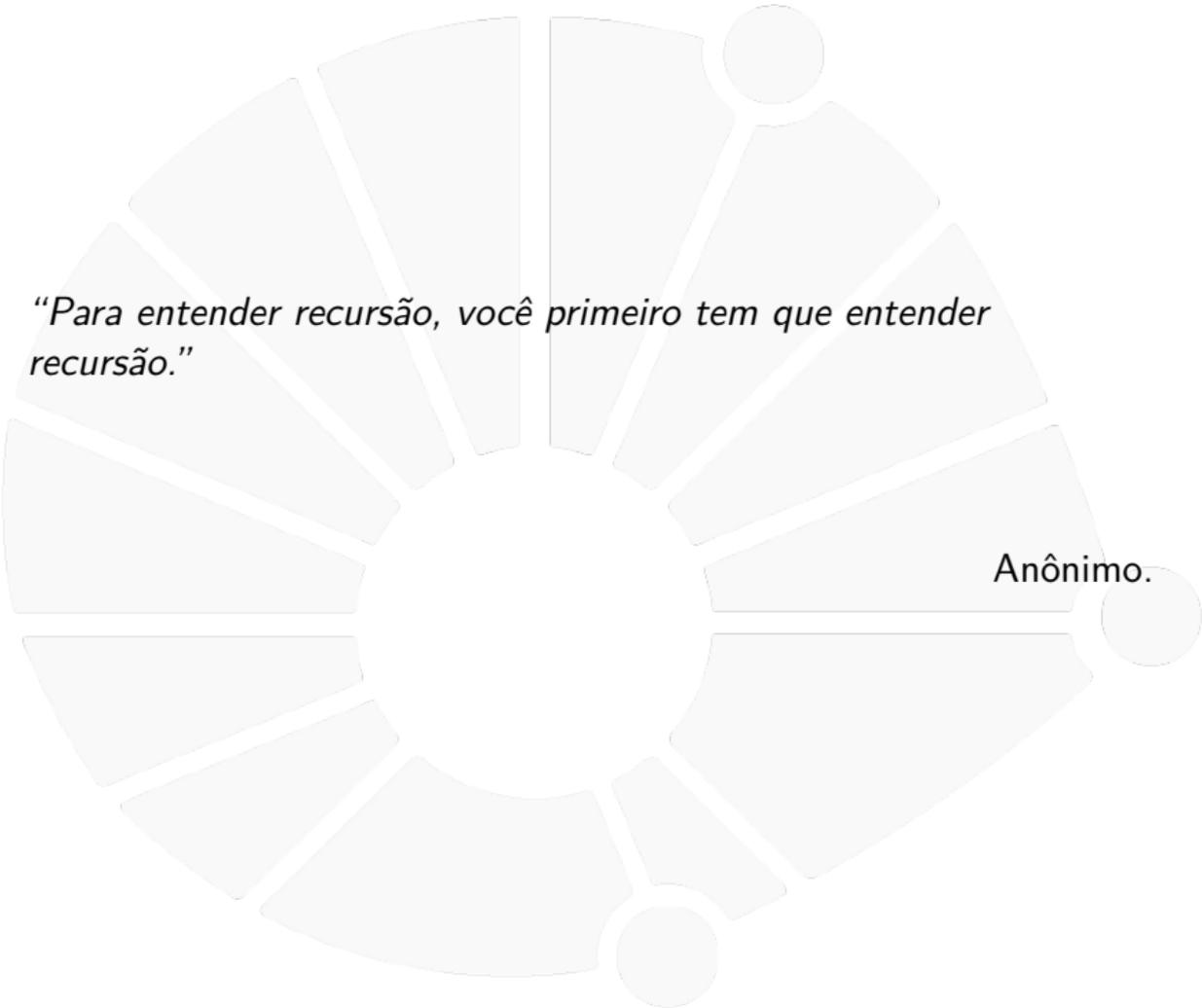
06/25

23



UNICAMP



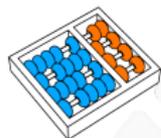


“Para entender recursão, você primeiro tem que entender recursão.”

Anônimo.



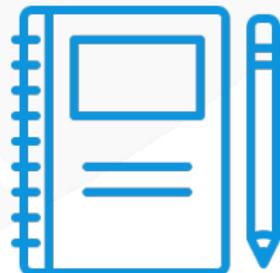
EXERCÍCIOS

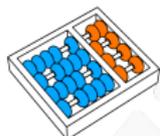


Recursão



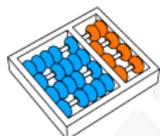
Soluções para os exercícios!





Exercícios

1. Faça uma função recursiva que calcula a soma dos números naturais menores ou iguais a n .
2. Faça uma função recursiva que calcula a soma dos números naturais ímpares menores ou iguais a n .
3. Faça uma função recursiva que calcula a soma de uma PA com valor inicial a_1 , razão r e n termos.
4. Faça uma função recursiva para contar quantos dígitos um número inteiro positivo tem na representação decimal.
5. Faça uma função recursiva que, dada uma string representando um número inteiro positivo em binário, acha o seu valor em decimal.
6. Faça uma função recursiva que, dado um número inteiro positivo, acha o seu valor em binário (em uma string).



Soluções

Soma até n :

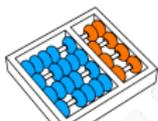
```
1 def soma(n):
2     if n <= 1:
3         return n
4     return n + soma(n - 1)
```

Soma de ímpares até n :

```
1 def soma_impar(n):
2     if n <= 1 :
3         return n
4     if n % 2 == 0:
5         return soma_impar(n - 1)
6     return n + soma_impar(n - 2)
```

Soma PA:

```
1 def soma_pa(a1, r, n):
2     if n == 1 :
3         return a1
4     return a1 + r * (n - 1) + soma_pa(n - 1)
```



Soluções. Continuação

Conta dígitos:

```

1 def conta_digitos(n):
2     if n < 10:
3         return 1
4     return 1 + conta_digitos(n // 10)

```

Binário a decimal:

```

1 def bin_dec_rec(s, k):
2     digito = ord(s[k]) - ord('0') # pega o digito atual se for 0 ou 1
3     if k == 0:
4         return digito
5     return 2 * bin_dec_rec(s, k - 1) + digito
6
7 def bin_dec(s):
8     return bin_dec_rec(s, len(s) - 1)

```

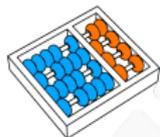
Decimal a binário:

```

1 def dec_bin(n):
2     if n < 2: # se for 0 ou 1, é só retornar o valor em str
3         return str(n)
4     return dec_bin(n // 2) + str(n % 2)

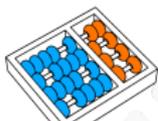
```

Se n (em decimal) tem $k + 1$ dígitos binários (onde cada dígito o denotamos por d_i , com i entre 0 e k), então:
 $n = 2^k \cdot d_0 + 2^{k-1} \cdot d_1 + \dots + 2^1 \cdot d_{k-1} + 2^0 \cdot d_k = 2(2(d_0 \dots) + d_{k-1}) + d_k$. Ou seja o último dígito binário é o resto da divisão de n por 2, o penúltimo o resto da divisão de $\lfloor \frac{n}{2} \rfloor$ por 2 e assim por diante.



Exercícios com listas

1. Faça uma função recursiva que calcula a soma dos elementos de uma lista.
2. Faça uma função recursiva que encontra o máximo de uma lista.
3. Faça uma função recursiva que busca um elemento em uma lista não ordenada.
4. Faça uma função recursiva que recebe uma lista e devolve uma cópia da lista invertida.
5. Faça uma função recursiva que checa se duas listas dadas são iguais.



Soluções

Soma dos elementos:

```
1 def soma_rec(l, n):
2     if n == len(l):
3         return 0
4     return l[n] + soma_rec(l, n + 1)
5
6 def soma(l):
7     return soma_rec(l, 0)
```

Máximo:

```
1 def maximo_rec(l, n):
2     if n == 0:
3         return l[0]
4     return max(l[n], maximo_rec(l, n - 1))
5
6 def maximo(l):
7     return maximo_rec(l, len(l) - 1)
```

Busca:

```
1 def busca_rec(l, n, x):
2     if n < 0:
3         return -1
4     if l[n] == x:
5         return n
6     return busca_rec(l, n - 1, x)
7
8 def busca(l, x):
9     return busca_rec(l, len(l) - 1, x)
```



Soluções. Continuação

Cópia invertida:

```
1 def invertida(l, n):
2     if n == len(l):
3         return []
4     return invertida(l, n + 1) + [l[n]]
5
6 def invertida(l):
7     return invertida(l, 0)
```

Iguais:

```
1 def iguais_rec(l1, l2, n):
2     if n == 0:
3         return l1[0] == l2[0]
4     return l1[n] == l2[n] and iguais_rec(l1, l2, n - 1)
5
6 def iguais(l1, l2):
7     return len(l1) == len(l2) and iguais_rec(l1, l2, len(l1) - 1)
```

RECURSÃO. SOLUÇÕES AOS EXERCÍCIOS

Santiago Valdés Ravelo
[https://ic.unicamp.br/~santiago/
ravelo@unicamp.br](https://ic.unicamp.br/~santiago/ravelo@unicamp.br)

MC102 - Algoritmos e
Programação de
Computadores

06/25

23



UNICAMP

