

19 Reunião 19: 03/NOV/2020

Ideias

Pensar



Figure 1: Fonte: <https://www.shutterstock.com/>

19.1 Reuniões passadas

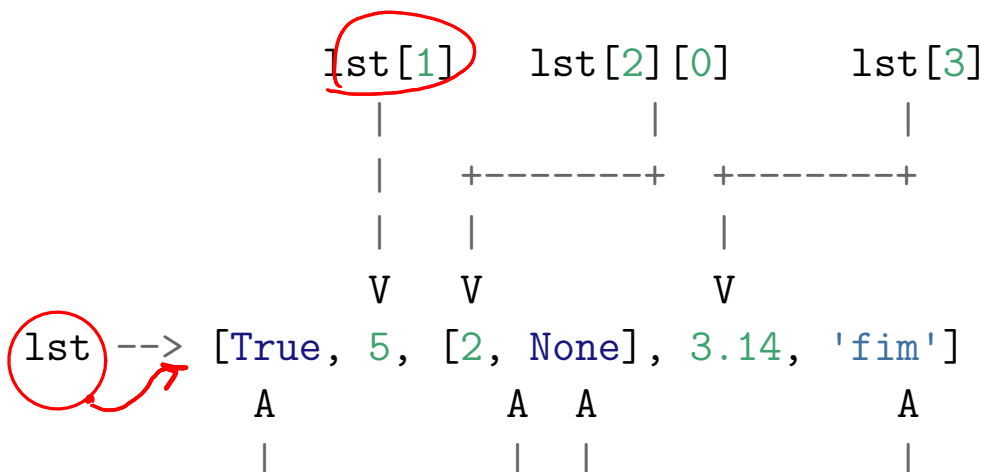
19.1.1 Comando for ... in range() e listas

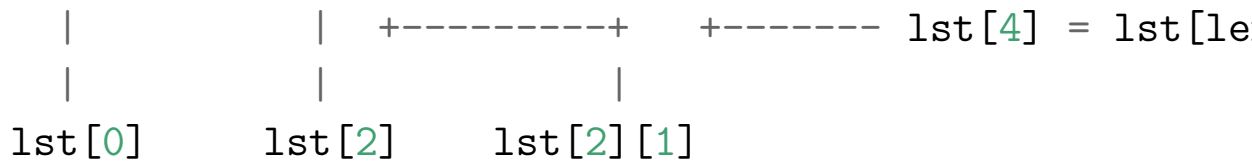
Considere a lista

```
lst = [True, 5, [2, None], 3.14, 'fim']
```

fim

ilustração dessa lista é





```
len(lista)      = 4
len(lista[2])  = 2
```

```
# criar uma lista
```

```
lst = []
```

```
for i in range(0, 6, 1):
```

```
    x = ...
```

```
    lst = lst + [x] # concatenação
```

```
# criar uma lista
```

```
uns = 5 * [1] # uns = [1, 1, 1, 1, 1]
```

```
# criar uma lista
```

```
primos = [2, 3, 5, 7, 11, 13, 17]
```

```
# percorre do início para o fim
```

```
n = len(lst)
```

```
for i in range(0, n, 1):
```

```
    print(i, ":", lst[i])
```

```
# percorre do fim para o início
```

```
for i in range(len(lst)-1, -1, -1):
```

```
    print(i, ":", lst[i])
```

Se não especificar, início = 0

Se não especificar, passo = 1

```
for i in range(10): # (fim)
```

```
print(i, end="")
```

Imprime: 0 1 2 3 4 5 6 7 8 9

19.1.2 Operador in list

O operador `in` pode ser usado para verificar se um item esta na lista

```
item in lst
```

é `True` se `item` é um elemento da lista `lst` e `False` em caso contrário. Por exemplo

```
5 in [2, 3, 5, 7, 11] == True
True in [True] == True
True in [False] == False
True in [1, 2, 5, None] == False
None in [1, 2, 5, None] == True
```

19.2 Hoje

- fatias
- apelidos versus clones
- operador `is`

19.3 Exercício 1: fatia de soma máxima

Uma **fatia** de uma lista é uma sublista especificada por um par de índices **ini** e **fim** que indicam início de fim (*exclusive*) da sublista. Por exemplo, para a lista

```
#indices  [-1, 23, 'oi', True, 2.71, None, 45]
           0   1   2       3     4     5   6
```

a fatia de:

- 2 a 5 é a sublista ['oi', True, 2.71]
- 3 a 7 é a sublista [True, 2.71, None, 45]
- 0 a 1 é a sublista [-1]
- 4 a 4 é a sublista vazia

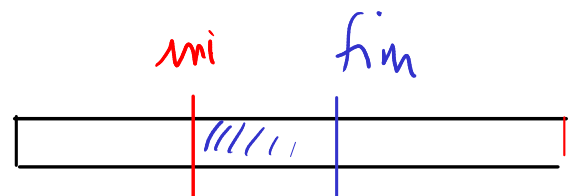
for i in range(**ini**, **fim**, **passo**)

ini **fim**

Dados **n > 0** e uma sequência com de **n números inteiros**, determinar uma fatia de soma máxima da lista formada pelos números da sequência.

19.3.1 Exemplos

```
Digite n: 12
Digite o 1o. número: 5
Digite o 2o. número: 2
Digite o 3o. número: -2
Digite o 4o. número: -7
Digite o 5o. número: 3
Digite o 6o. número: 14
Digite o 7o. número: 10
Digite o 8o. número: -3
Digite o 9o. número: 9
Digite o 10o. número: -6
Digite o 11o. número: 4
Digite o 12o. número: 1
```



```
lista = [5, 2, -2, -7, 3, 14, 10, -3, 9, -6, 4, 1]
fatia de soma máxima = [3, 14, 10, -3, 9]
```

soma=33

ini = 4

fim = 9

Digite n: 5
Digite o 1o. número: 1
Digite o 2o. número: 2
Digite o 3o. número: -3
Digite o 4o. número: 4
Digite o 5o. número: 5
lista = [1, 2, -3, 4, 5]
fatia de soma máxima= [1, 2, -3, 4, 5]
soma= 9
ini = 0
fim = 5

Digite n: 5
Digite o 1o. número: -1
Digite o 2o. número: 2
Digite o 3o. número: -3
Digite o 4o. número: 4
Digite o 5o. número: 5
lista = [-1, 2, -3, 4, 5]
fatia de soma máxima= [4, 5]
soma= 9
ini = 3
fim = 5

Digite n: 5
Digite o 1o. número: 1
Digite o 2o. número: -2
Digite o 3o. número: 3
Digite o 4o. número: -4
Digite o 5o. número: 5
lista = [1, -2, 3, -4, 5]
fatia de soma máxima= [5]
soma= 5

$s = lst[0]$
 $ini = 0$
 $fin = 1$
return s, ini, fin

19.3.2 Proposta de solução

- (a) Escreva função `fatia()` que **recebe** uma lista `lst` e índices `ini` e `fim` e **retorna** uma lista formada pela sublista de `lst` com índices entre `ini` e `fim`. Por exemplo,

```
In [12]: lst = [1, 'oi', True, 3.14, 19, None]
```

```
In [13]: fatia(lst, 1, 4)
Out[13]: ['oi', True, 3.14]
```

```
In [14]: fatia(lst, 2, len(lst))
Out[14]: [True, 3.14, 19, None]
'''
```

- (b) Escreva uma função `soma()` que **recebe** um lista de números e retorna a soma de seus elementos. Por exemplo,

```
In [15]: soma([1,2,3])
Out[15]: 6
```

```
In [16]: soma([-1,2,-3])
Out[16]: -2
```

- (c) Escreva uma função `fatia_max()` que **recebe** uma lista de números inteiros e **retorna** o índices do início e do fim de uma fatia de soma máxima e o valor da soma máxima. Por exemplo,

```
In [18]: fatia_max([1,-2, 3, -2, 5])
Out[18]: (2, 5, 6)
```

```
In [19]: fatia_max([10,-2,-5,-2, 10])
Out[19]: (0, 5, 11)
```

- (d) Escreva uma função `leia_seq()` que **lê** um inteiro positivo `n` e uma sequência de `n` números inteiros e retorna a lista formada por esses números.

- (e) Escreva uma função `main()` que lê um inteiro $n > 0$ e uma seqüência n números inteiros e calcula a maior soma de uma de suas fatias.

19.3.3 Solução

```
def main():
    '''(None) -> None
    Programa que lê  $n > 0$  e uma seqüência  $n$ 
    números inteiros e calcula a maior soma de
    uma de suas fatias.
    '''
    lst = leia_seq()
    print("lista =", lst)
    ini, fim, soma = fatia_max(lst)
    print("fatia de soma máxima=", fatia(lst, ini, fim))
    print("soma=", soma)
    print("ini =", ini)
    print("fim =", fim)

#-----
def leia_seq():
    '''(None) -> list
    Função que lê um número inteiro positivo `n` e uma seqüência
    com `n` números inteiros.
    RETORNA a lista com os números lidos.
    '''
    lst = []
    n = int(input("Digite n: "))
    for i in range(n):
        num = int(input("Digite o %do. número: %(i+1)))
        lst += [num]
    return lst
```

#-----

```
def fatia_max(lst):  
    '''(list) -> int, int, int
```

*RECEBE uma lista de números inteiros.
RETORNA o índices do início e do fim
de uma fatia de soma máxima e o valor
da soma máxima*

```
In [18]: fatia_max([1,-2, 3, -2, 5])  
Out[18]: (2, 5, 6)
```

```
In [19]: fatia_max([10,-2,-5,-2, 10])  
Out[19]: (0, 5, 11)  
'''
```

```
n = len(lst)  
if n == 0: return 0, 0, None  
smax = lst[0]  
ini = 0  
fim = 1  
for i in range(0, n, 1):  
    for j in range(i+1, n+1, 1):  
        sublista = fatia(lst, i, j)  
        s = soma(sublista)  
        if s > smax:  
            smax = s  
            ini = i  
            fim = j  
return ini, fim, smax
```

aberto
atenção!
n+1 pois desejamos que j assumma o valor n

#-----

```
def soma(lst):
```

```
'''(list) -> int
```

RECEBE uma lista `lst`.

RETORNA a soma dos valores na lista.

```
In [15]: soma([1,2,3])
```

```
Out[15]: 6
```

```
In [16]: soma([-1,2,-3])
```

```
Out[16]: -2
```

```
'''
```

```
soma = 0
```

```
n = len(lst)
```

```
for i in range(0, n, 1):
```

```
    soma += lst[i]
```

```
return soma
```

```
def fatia(lst, ini, fim):
```

```
    '''(list, int, int) -> lst
```

RECEBE uma lista `lst` e indices `ini` e `fim` e retorna uma lista forma pela sublista de lst com indices entre ini e fim.

```
In [12]: lst = [1, 'oi', True, 3.14, 19, None]
```

```
In [13]: fatia(lst, 1, 4)
```

```
Out[13]: ['oi', True, 3.14]
```

```
In [14]: fatia(lst, 2, len(lst))
```

```
Out[14]: [True, 3.14, 19, None]
```

```
'''
```

```
sublista = []
```

```
for i in range(ini, fim, 1):
```

```
    sublista = sublista + [lst[i]]  
    return sublista
```

```
# inicio do programa
```

```
main()
```

|

19.4 Fatias em Python

Em Python uma fatia de uma lista é especificada por um intervalo de índices

```
lista[início:fim] # clone_lista(lista, início, fim)
```

e é um clone ou sublista.

super importante essa coisa de fatia

Podemos usar `:` para definir qualquer *fatia* de uma lista. Por exemplo,

```
lista1[1:3] retorna a lista [1, 2]
```

```
lista1[:] é uma abreviatura de lista1[início:len(lista1)]
```

```
def main():
```

```
    lista1 = [0, 1, 2, 3, 4]
```

```
    lista2 = lista1[:] # faz uma clone da lista
```

```
    lista2[1] = 7
```

```
    print ("lista1 = ", lista1)
```

```
    print ("lista2 = ", lista2)
```

*veremos / discutiremos
exemplos no visualizador*

```
main()
```

19.4.1 Solução *de luxe*

```
def main():
    '''(None) -> None
    Programa que lê  $n > 0$  e uma seqüência  $n$ 
    números inteiros e calcula a maior soma de
    um de seus segmento.
    '''
    lst = leia_seq()
    print("lista =", lst)
    ini, fim, soma = fatia_max(lst)
    print("fatia de soma máxima=", lst[ini:fim])
    print("soma=", soma)
    print("ini =", ini)
    print("fim =", fim)
```

```
#-----
def leia_seq():
    '''(None) -> list

    Função que lê um número inteiro positivo `n` e uma seqüência
    com `n` números inteiros.
    RETORNA a lista com os números lidos.
    '''
    lst = []
    n = int(input("Digite n: "))
    for i in range(n):
        num = int(input("Digite o %do. número: %(i+1)))
        lst += [num]
    return lst
```

```
#-----
def fatia_max(lst):
```



```
'''(list) -> int, int, int
```

RECEBE uma lista de números inteiros.

RETORNA o índices do início e do fim de uma fatia de soma máxima e o valor da soma máxima

```
'''
```

```
n = len(lst)
```

```
if n == 0: return 0, 0, None
```

```
smax = lst[0]
```

```
ini = 0
```

```
fim = 1
```

```
for i in range(0, n, 1):
```

```
    for j in range(i+1, n+1, 1):
```

```
        s = sum(lst[i:j]) # sum() é uma função nativa
```

```
        if s > smax:
```

```
            smax = s
```

```
            ini = i
```

```
            fim = j
```

```
return ini, fim, smax
```

função que soma os elementos de uma lista

fatia do Python

```
# inicio do programa
```

```
main()
```