

#### Algoritmos e Estrutura de Dados

Recursividade

Prof. Dr. Diego Rodrigues de Almeida diego.rodrigues@afogados.ifpe.edu.br https://sites.google.com/site/ifpediego

# Recursividade



- Recursividade é o mecanismo de programação no qual uma definição de função ou de outro objeto refere-se ao próprio objeto sendo definido.
- Assim função recursiva é uma função que é definida em termos de si mesma.
- Recursividade é o mecanismo básico para repetições nas linguagens funcionais.
- ▶ São sinônimos: recursividade, recursão, recorrência



#### Estratégia recursiva

- Estratégia para a definição recursiva de uma função:
  - 1. Dividir o problema em problemas menores do mesmo tipo
  - 2. Resolver os problemas menores (dividindo-os em problemas ainda menores, se necessário)
  - Combinar as soluções dos problemas menores para formar a solução final
- Ao dividir o problema sucessivamente em problemas menores eventualmente os casos simples são alcançados:
  - Não podem ser mais divididos
  - Suas soluções são definidas explicitamente



## Definição recursiva

- De modo geral, uma definição de função recursiva é dividida em duas partes:
  - Há um ou mais casos base
    - Dizem o que fazer em situações simples, onde não é necessária nenhuma recursão.
    - Nestes casos a resposta pode ser dada de imediato, sem chamar recursivamente a função sendo definida. Isso garante que a recursão eventualmente possa parar.
  - Há um ou mais casos recursivos
    - São mais gerais
    - Definem a função em termos de uma chamada mais simples a si mesma.



- O fatorial de um número é um conceito matemático usado para denotar o produto de todos os números inteiros positivos de l até o próprio número.
- $\blacktriangleright$  É representado pelo símbolo n! onde n é o número para o qual você deseja calcular o fatorial.
- $\blacktriangleright$  A fórmula para calcular o fatorial de um número n é a seguinte:

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 2 \times 1$$

- Aqui estão alguns exemplos de fatoriais:
  - 0! = 1 (Por definição, o fatorial de 0 é definido como I.)
  - ▶ 1! = 1

  - $3! = 3 \times 2 \times 1 = 6$
  - $4! = 4 \times 3 \times 2 \times 1 = 24$
  - $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$





- Podemos dizer que o fatorial de um número é esse número vezes o fatorial do seu antecessor
- Exemplo:

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

$$4!$$

$$4! = 4 \times 3 \times 2 \times 1$$

$$3!$$

$$3! = 3 \times 2 \times 1$$

$$2! = 2 \times 1$$

$$1!$$

$$1! = 1 \times 0!$$

$$0! = 1$$





- Assim, podemos dizer que  $n! = n \times (n-1)!$ 
  - Esse é o nosso caso recursivo!!!
- Assim:
  - $n! = n \times (n-1)!$
  - $(n-1)! = (n-1) \times (n-2)!$
  - $(n-2)! = (n-2) \times (n-3)!$
  - $(n-3)! = (n-3) \times (n-4)!$
- Isso vai ter fim?
  - Precisamos do nosso caso base se não ficaremos nessa recursão infinitamente





- Qual é o nosso caso base?
  - 0! = 1
- Assim:
  - $n! = n \times (n-1)!$
  - $(n-1)! = (n-1) \times (n-2)!$
  - $(n-2)! = (n-2) \times (n-3)!$
  - $(n-3)! = (n-3) \times (n-4)!$
  - ...
  - 0! = 1
- VAMOS FAZER O CÓDIGO DE FATORIAL USANDO RECURSIVIDADE



A função que calcula o fatorial de um número natural pode ser definida recursivamente como segue:

```
def fatorial(n):
    if(n == 0):
        return 1
    return n * fatorial(n-1)
```

- A primeira equação estabelece que o fatorial de 0 é 1. Este é o caso base.
- A segunda equação estabelece que o fatorial de um número positivo é o produto deste número e do fatorial do seu antecessor. Este é o caso recursivo.
- Dbserve que no caso recursivo o subproblema fatorial (n-1) está mais próximo do caso base fatorial 0.





```
def fatorial(n):
    if(n == 0):
        return 1
    resultado = n * fatorial(n-1)
    return resultado

n = int(input('Digite um número: '))
fatorialN = fatorial(n)
print('O fatorial de ', n, ' é ', fatorialN)
```





```
def fatorial(n):
    if(n == 0):
        return 1
    resultado = n * fatorial(n-1)
    return resultado

n = int(input('Digite um número: '))
fatorialN = fatorial(n)
    print('O fatorial de ', n, ' é ', fatorialN)
```











```
n=4
def fatorial(n):
   if(n == 0):
      return 1
   resultado = n * fatorial(n-1)
   return resultado
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
                              resultado=5*?
    return resultado
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                           fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```



```
n=4
def fatorial(n):
   if(n == 0):
      return
   resultado = n * fatorial(n-1)
   return resultado
                  resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
                              resultado=5*?
    return resultado
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                           fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=3
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
               n=4
def fatorial(n):
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                              resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                            fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=3
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=3*?
               n=4
def fatorial(n):
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                              resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                            fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                   if(n == 0):
       return 1
                                                       return 1
   resultado = n * fatorial(n-1)
                                                   resultado = n * fatorial(n-1)
   return resultado
                                                   return resultado
                   resultado=3*?
               n=4
def fatorial(n):
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                               resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                             fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                    if(n == 0):
       return 1
                                                       return 1
   resultado = n * fatorial(n-1)
                                                    resultado = n * fatorial(n-1)
                                                    return resultado resultado=2*?
   return resultado
                   resultado=3*?
               n=4
def fatorial(n):
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                               resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                             fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
                                                      if(n == 0):
   if(n == 0):
       return 1
                                                          return 1
    resultado = n * fatorial(n-1)
                                                      resultado = n \stackrel{*}{\wedge} fatorial(n-1)
   return resultado
                                                      return resul
                                                                      resultado=2*?
                    resultado=3*?
                                                  def fatorial(n): n=|
                n=4
def fatorial(n):
   if(n == 0):
                                                      if(n == 0):
       return 1
                                                          return 1
                                                      resultado = n * fatorial(n-1)
   resultado = n * fatorial(n-1)
   return resultado
                                                      return resultado
                    resultado=4*?
def fatorial(n): n=5
     if(n == 0):
         return 1
     resultado = n * fatorial(n-1)
    return resultado
                                 resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                                fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                      if(n == 0):
       return 1
                                                          return 1
    resultado = n * fatorial(n-1)
                                                      resultado = n \stackrel{*}{\angle} fatorial(n-1)
   return resultado
                                                      return resul
                                                                      resultado=2*?
                    resultado=3*?
                                                   def fatorial(n): n=|
                n=4
def fatorial(n):
   if(n == 0):
                                                      if(n == 0):
       return 1
                                                          return
   resultado = n * fatorial(n-1)
                                                       resultado = n * fatorial(n-1)
   return resultado
                                                       return resultado
                    resultado=4*?
                                                                       resultado= 1*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                                 resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                                fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                       if(n == 0):
       return 1
                                                           return 1
    resultado = n * fatorial(n-1)
                                                       resultado = n \stackrel{*}{\angle} fatorial(n-1)
   return resultado
                                                       return resul
                                                                      resultado=2*?
                     resultado=3*?
                                                   def fatorial(n): n=1
                 n=4
def fatorial(n):
   if(n == 0):
                                                       if(n == 0):
       return 1
                                                           return 1
   resultado = n * fatorial(n-1)
                                                       resultado = n \wedge fatorial(n-1)
   return resultado
                                                       return resul
                     resultado=4*?
                                                                        resultado=I*?
def fatorial(n): n=5
                                                   def fatorial(n):
                                                                   n=0
                                                       if(n == 0):
    if(n == 0):
                                                           return 1
          return 1
                                                       resultado = n * fatorial(n-1)
     resultado = n * fatorial(n-1)
                                                       return resultado
    return resultado
                                 resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                                fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                       if(n == 0):
       return 1
                                                           return 1
    resultado = n * fatorial(n-1)
                                                       resultado = n \stackrel{*}{\angle} fatorial(n-1)
   return resultado
                                                       return resul
                                                                       resultado=2*?
                     resultado=3*?
                                                   def fatorial(n): n=1
                 n=4
def fatorial(n):
   if(n == 0):
                                                       if(n == 0):
       return 1
                                                           return 1
   resultado = n * fatorial(n-1)
                                                       resultado = n \wedge fatorial(n-1)
   return resultado
                                                       return resul
                    resultado=4*?
                                                                        resultad
def fatorial(n): n=5
                                                   def fatorial(n);
                                                                   n=0
                                                       if(n == 0):
     if(n == 0):
                                                           return 1
          return 1
                                                       resultado = n * fatoria.
     resultado = n * fatorial(n-1)
                                                       return resultado
    return resultado
                                 resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                                fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                      if(n == 0):
       return 1
                                                          return 1
    resultado = n * fatorial(n-1)
                                                      resultado = n \stackrel{*}{\wedge} fatorial(n-1)
   return resultado
                                                      return resul
                                                                      resultado=2*?
                    resultado=3*?
                                                  def fatorial(n): n=1
                n=4
def fatorial(n):
   if(n == 0):
                                                      if(n == 0):
       return 1
                                                          return 1
   resultado = n * fatorial(n-1)
                                                      resultado = n
   return resultado
                                                      return resultado
                    resultado=4*?
                                                                       resultado=1*1
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                                 resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                                fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                      if(n == 0):
       return 1
                                                          return 1
    resultado = n * fatorial(n-1)
                                                      resultado = n \stackrel{*}{\angle} fatorial(n-1)
   return resultado
                                                      return resul
                                                                      resultado=2*?
                    resultado=3*?
                                                   def fatorial(n): n=1
                n=4
def fatorial(n):
   if(n == 0):
                                                      if(n == 0):
       return 1
                                                          return 1
   resultado = n * fatorial(n-1)
                                                      resultado = n * fatorial(n-1)
   return resultado
                                                       return resultado
                    resultado=4*?
                                                                        resultado= I
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                                 resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                                fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                      if(n == 0):
                                                                       resultado=2*1
       return 1
                                                          return 1
    resultado = n * fatorial(n-1)
                                                      resultado = n \stackrel{*}{\sim} fatorial(n-1)
   return resultado
                                                      return resul/
                    resultado=3*?
                n=4
                                                  def fatorial(n): n=1
def fatorial(n):
   if(n == 0):
                                                      if(n == 0):
       return 1
                                                          return 1
                                                      resultado = n * fatorial(n-1)
   resultado = n * fatorial(n-1)
   return resultado
                                                      return resultado
                    resultado=4*?
def fatorial(n): n=5
     if(n == 0):
         return 1
     resultado = n * fatorial(n-1)
    return resultado
                                 resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                                fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
   if(n == 0):
                                                   if(n == 0):
                                                                   resultado=2*1
       return 1
                                                       return 1
   resultado = n * fatorial(n-1)
                                                   resultado = n *
   return resultado
                                                   return resultado
                   resultado=3*?
                n=4
def fatorial(n):
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                               resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                             fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=2
def fatorial(n): n=3
                                                    if(n == 0):
   if(n == 0):
                                                                    resultado=2
       return 1
                                                       return 1
   resultado = n * fatorial(n-1)
                                                   resultado = n * fatorial(n-1)
   return resultado
                                                   return resultado
                   resultado=3*?
               n=4
def fatorial(n):
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                               resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                             fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=3
   if(n == 0):
       return 1
   resultado = n *
   return resultado
                   resultado=3*2
               n=4
def fatorial(n):
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
                              resultado=5*?
    return resultado
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                            fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```



```
def fatorial(n): n=3
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                    resultado=6
               n=4
def fatorial(n):
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
                              resultado=5*?
    return resultado
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                            fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

```
def fatorial(n): n=3
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=6
def fatorial(n):
               n=4
   if(n == 0):
       return 1
   resultado = n * fatorial(n-1)
   return resultado
                   resultado=4*?
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
                              resultado=5*?
    return resultado
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                            fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```



```
n=4
def fatorial(n):
   if(n == 0):
      return 1
   resultado = n *
   return resultado
                   resultado=4*6
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
                              resultado=5*?
    return resultado
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                           fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```



```
n=4
def fatorial(n):
   if(n == 0):
      return 1
   resultado = n * fatorial(n-1)
   return resultado
                    resultado=24
def fatorial(n): n=5
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                              resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                           fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```



```
n=4
def fatorial(n):
   if(n == 0):
      return 1
   resultado = n * fatorial(n-1)
   return resultado
                    resultado=24
def fatorial(n).
    if(n == 0):
         return 1
    resultado = n * fatorial(n-1)
    return resultado
                              resultado=5*?
n = int(input('Digite um número: '))
fatorialN = fatorial(n)
                                            fatorialN=?
print('O fatorial de ', n, ' é ', fatorialN)
```

























Implemente uma função recursiva chamada somaAteN que recebe um número inteiro positivo n e retorna a soma de todos os números de I até n.

- Por exemplo:
  - $\triangleright$  somaAteN(3) deve retornar 6, pois I + 2 + 3 = 6.





 A sequência de Fibonacci é a sequência numérica proposta pelo matemático Leonardo Pisa, mais conhecido como Fibonacci

 A sequência começa por 0 e 1, a partir de então, o próximo número é sempre a soma dos dois números que o antecedem

$$F_n = F_{n-2} + F_{n-1}$$

Sequência de Fibonacci: 0, I, I, 2, 3, 5, 8, 13, 21, ...



#### Exemplo: Fibonacci

Resolução I

```
def fib(n):
    if(n == 1):
        return 0
    if(n == 2):
        return 1
    return fib(n-2) + fib(n-1)

n = int(input('Digite um número: '))
print('Os ', n, ' números da sequência de Fibonacci são: ', end='')
for i in range(1, n + 1):
    print(fib(i), ' ', end='')
```



#### Exemplo: Fibonacci

#### Resolução 2

```
def fibonacci(n):
    if n \le 0:
        return []
    elif n == 1:
        return [0]
    elif n == 2:
        return [0, 1]
    else:
        # Chamada recursiva para obter os "n-1" números anteriores
        fib = fibonacci(n - 1)
        # Calcula o próximo número de Fibonacci como a soma dos dois anteriores
        fib.append(fib[-1] + fib[-2])
        return fib
n = int(input('Digite um número: '))
print('Os ', n, ' números da sequência de Fibonacci são: ', end='')
print(fibonacci(n))
```



#### Exemplo: Busca Linear Recursiva

```
def buscaLinear(vetor, tamanho, n):
    if(tamanho == 0):
        return -1
    if(vetor[tamanho - 1] == n):
        return tamanho - 1
    return buscaLinear(vetor, tamanho - 1, n)
vetor = [1, 5, 8, 3, 14, 4, 12, 88, 18]
n = int(input('Digite um número: '))
posicao = buscaLinear(vetor, len(vetor), n)
if(posicao < 0):</pre>
    print('O número ', n, ' não se encontra no vetor')
else:
    print('O número ', n, ' está na posicao ', posicao)
```

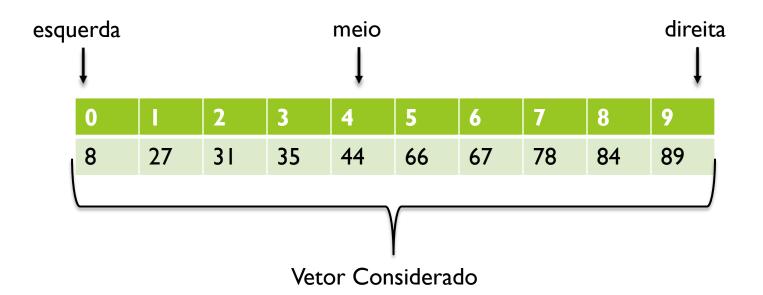


 Vamos ver agora a implementação do algoritmo de busca binária de forma recursiva

Vamos primeiro relembrar de como ele funciona

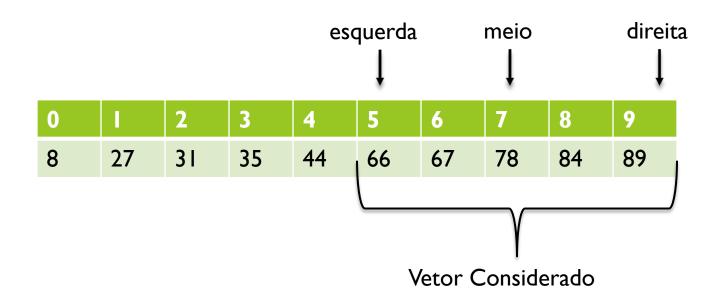


Procurar por 66 66 > 44



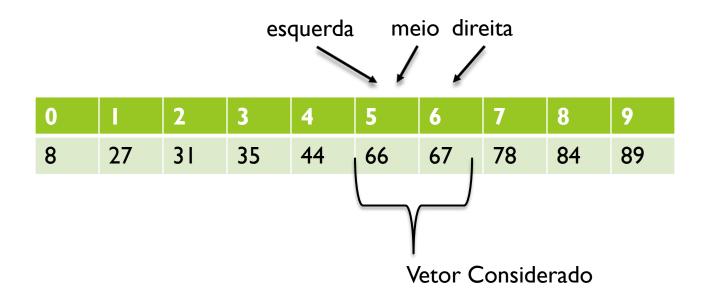


Procurar por 66 66 < 78



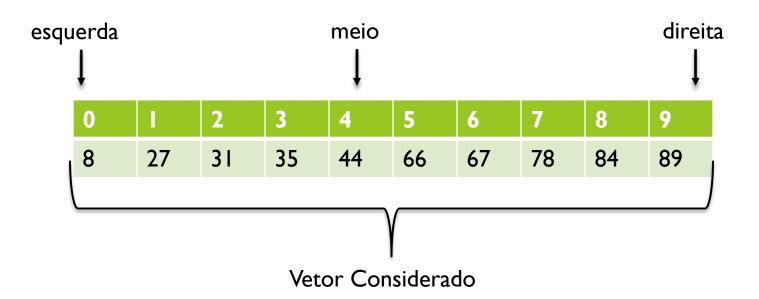


Procurar por 66



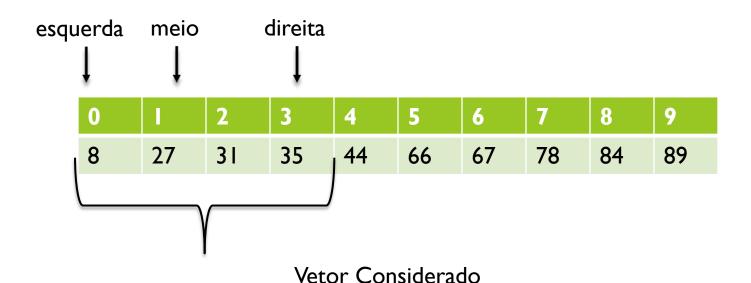


Procurar por 31 31 < 44



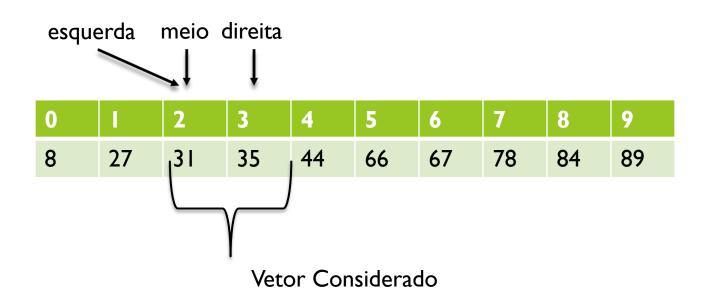


Procurar por 31 > 27





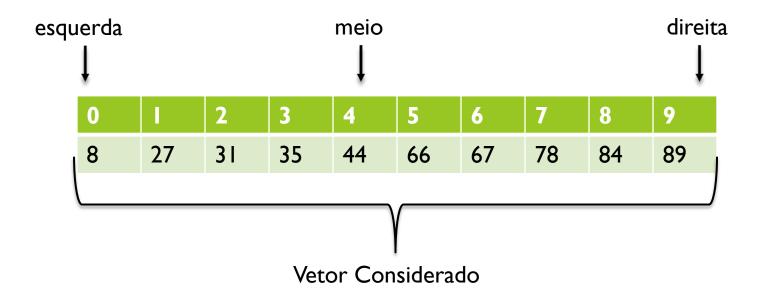
Procurar por 31 = 31 Achou!!!





Procurar por 80

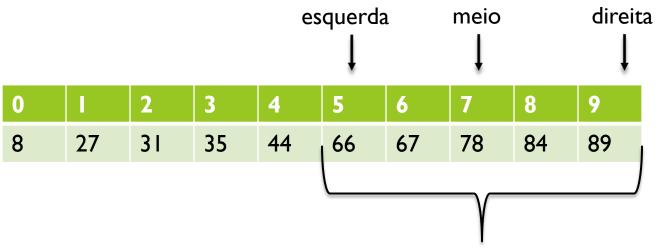
80 > 44





Procurar por 80

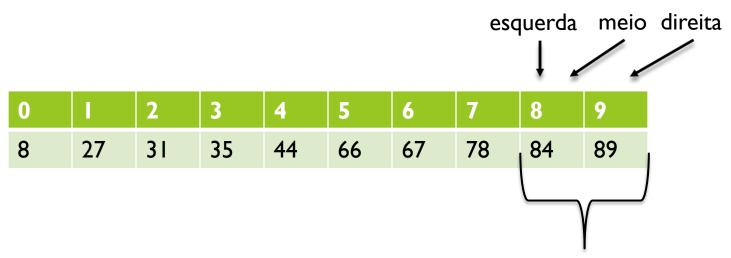
80 > 78





Procurar por 80

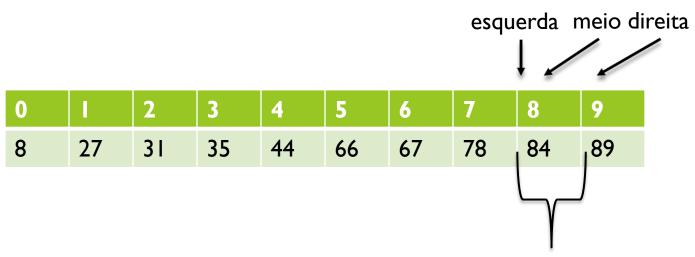
80 < 84





Procurar por 80

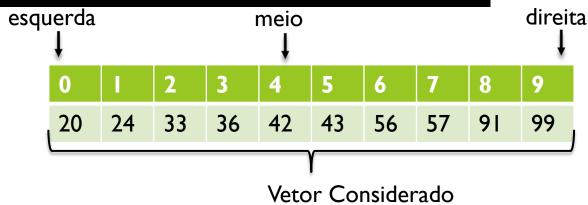
80 ≠ 84 Não achou!!!





```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d)
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

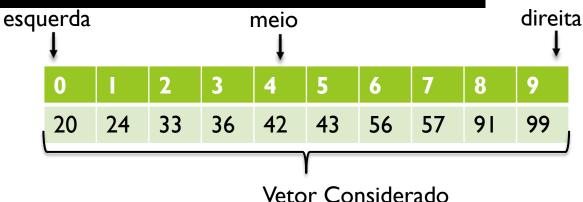
Variável	Valor
е	0
d	9
meio	
×	99





```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d)
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

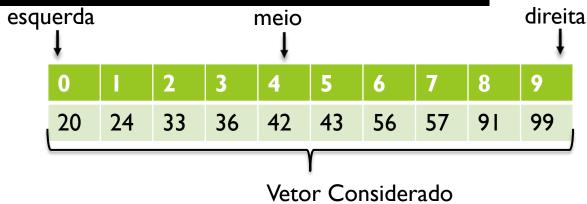
Variável	Valor
е	0
d	9
meio	4
X	99





```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d)
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	0
d	9
meio	4
x	99

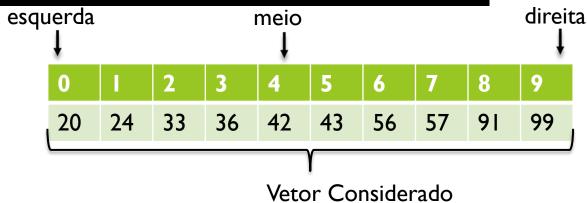




```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio

if (e == d):
    return -1 #Não foi encontrado
elif (vetor[meio] < x):
    return buscaBinaria(x, vetor, meio+1, d)
else:
    return buscaBinaria(x, vetor, e, meio-1)</pre>
```

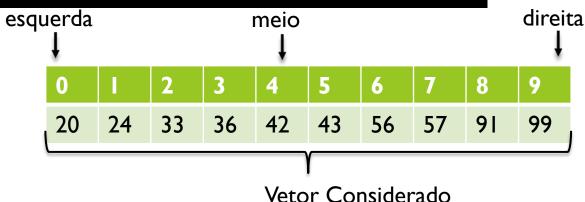
Variável	Valor
е	0
d	9
meio	4
×	99





```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d)
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

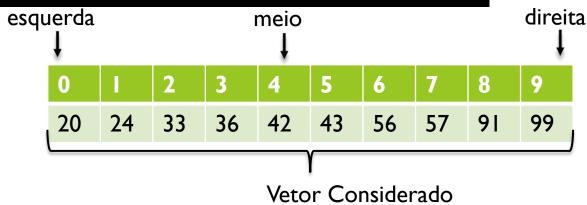
Variável	Valor
е	0
d	9
meio	4
×	99





```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	0
d	9
meio	4
x	99





```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	5
d	9
meio	7
x	99

				es	squer ↓	da r	neio ↓		direit Ļ
0	1	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
Vetor Considerado									



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	5
d	9
meio	7
x	99

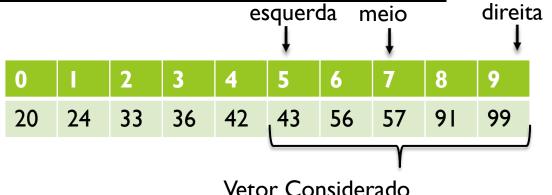
				es	squer   	da r	neio ↓		direi
0	ı	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
	Vetor Considerado								



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio

if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	5
d	9
meio	7
x	99





```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) la chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	5
d	9
meio	7
x	99

				es	squer ↓	da r	neio ↓		direi Ļ
0	ı	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
				Veto	or Co	nside	γ rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | e 2 chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	5
d	9
meio	7
X	99

				es	squer ↓	da n	neio ↓		direit ↓	
0	1	2	3	4	5	6	7	8	9	
20	24	33	36	42	43	56	57	91	99	
	20 24 33 36 42 43 56 57 91 99  Vetor Considerado									



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | e 2 chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	8
d	9
meio	8
X	99

						esque	erda	meio ↓↓	direit:
0	1	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
							rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) la e 2a chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	8
d	9
meio	8
x	99

						esque	erda	meio ↓↓	direit ↓
0	ı	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
				Veto	r Co	nside	rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio

if (e == d):
        return -1 #Não foi encontrado
elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | a e 2a chamada
else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	8
d	9
meio	8
x	99

						esque	erda	meio ↓↓	direita ↓
0	ı	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
				Veto	or Co	nside	rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) la e 2a chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	8
d	9
meio	8
x	99

						esque	erda	meio ↓↓	direit
0	I	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
									γ—
				Veto	or Co	nside	rado		



Variável	Valor
е	8
d	9
meio	8
x	99

						esque	erda	meio ↓ ↓	direit ↓
0	1	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
				Veto	r Co	nside	rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | a, 2a e 3a chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	9
d	9
meio	9
x	99

						esque	erda	meio	direit
0	1	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91,	99
									$\gamma$
				Veto	or Co	nside	rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | a, 2a e 3a chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	9
d	9
meio	9
x	99

						esque	erda	meio	direit
0	1	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91,	99
									~
				Voto	or Co	nsida	rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio

    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | a, 2a e 3a chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	9
d	9
meio	9
×	99

						esque	erda	meio	direit
0	ı	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91,	99
									~
				Veto	or Co	nside	rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | a e 2a chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	9
d	9
meio	9
×	99

						esque	erda	meio	direit
0	1	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91,	99
									~
				Voto	or Co	nsida	rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d) | chamada
    else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	9
d	9
meio	9
×	99

						esque	erda	meio	direit:
0	1	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91,	99
								,	$\mathcal{T}$
				Veto	or Co	nside	rado		



```
# x => chave | v[] => vetor ordenado |
# e => Limite inferior (esquerda) | d => Limite
Superior (direita)
def buscaBinaria(x, vetor, e, d):
    meio = int((e + d)/2)
    if (vetor[meio] == x):
        return meio
    if (e == d):
        return -1 #Não foi encontrado
    elif (vetor[meio] < x):
        return buscaBinaria(x, vetor, meio+1, d)
else:
        return buscaBinaria(x, vetor, e, meio-1)</pre>
```

Variável	Valor
е	9
d	9
meio	9
x	99

						esque	erda	meio	direit:
0	ı	2	3	4	5	6	7	8	9
20	24	33	36	42	43	56	57	91	99
									$\mathcal{T}$
	Vetor Considerado								



#### Bibliografia

- Lintzmayer, Carla Negri; Mota, Guilherme Oliveira. Análise de Algoritmos e Estrutura de Dados.
- Cormen, Thomas. et al. Algoritmos Teoria e Prática: tradução.
   2º Edição. Elsevier.



#### Pense nisso...

- Persista! A caminhada é longa, mas o destino vale cada esforço."
  - Autor desconhecido

