

# Algoritmos 2025/1 BCC

<b>Conhecimentos básicos em Informática (31/05/2025)</b>	<b>1</b>
<b>Beecrowd (27/02/2025)</b>	<b>1</b>
<b>Sites de referência das linguagens (06/03/2025)</b>	<b>1</b>
<b>Tipos de Dados (20/03/2025)</b>	<b>2</b>
<b>Problemas (27/02/2025)</b>	<b>2</b>
1) Média (27/02/2025)	3
2) Ordenação (13/03/2025)	4
3) Leitura de arquivos (27/03/2025)	8
<b>4) Recursividade (02/04/2025)</b>	<b>9</b>
5) Gráficos (12/06/2025)	11
<b>Exercícios selecionados do BeeCrowd (02/04/2025)</b>	<b>11</b>
<b>Exercícios diversos (19/03/2025)</b>	<b>12</b>
<b>Exercícios preparatórios para a AV1 (23/04/2025)</b>	<b>12</b>



## Conhecimentos básicos em Informática (31/05/2025)

<https://docs.google.com/document/d/1scm9OUQ08TuBRXCrQpsk17CTjMY3CJLK9OjLq26c5vo/edit?usp=sharing>

## Beecrowd (27/02/2025)

- <https://judge.beecrowd.com/>
- Plataforma para resolver problemas
- Treinamento de lógica de programação
- Treinamento para competições de programação
- O cadastro é demorado e trabalhoso
- Quem é o mais forte do IFC?
  - <https://judge.beecrowd.com/en/users/university/ifc-geral>

## Sites de referência das linguagens (06/03/2025)

python e C:

<https://www.w3schools.com/>

python:

<https://www.redhat.com/en/services/training/ad141-red-hat-training-presents-introduction-to-python-programming>

<https://www.netacad.com/courses/python-essentials-1>

<https://docs.python.org/3/tutorial/index.html>

C:

<https://www.programiz.com/c-programming>

<https://www.learn-c.org/>

<https://beej.us/guide/bgc/html/>

Contribuições do **Matheus** (obrigado!):

- Uma outra plataforma interessante de estudo de algoritmos é o Scratch, desenvolvido pelo MIT. Segue o link da plataforma: <https://scratch.mit.edu/>
- Existe um curso gratuito de Harvard chamado CS50. Cada aula é extensa e tem geralmente 1h+ explicando conceitos da Ciência da Computação. Segue o link para o material do curso: <https://cs50.harvard.edu/x/2025/weeks/0/>

## Tipos de Dados (20/03/2025)

[https://www.w3schools.com/c/c\\_data\\_types.php](https://www.w3schools.com/c/c_data_types.php)

[https://www.w3schools.com/python/python\\_datatypes.asp](https://www.w3schools.com/python/python_datatypes.asp)

tipo	python	c
texto	str	char
número	int float complex	int float double
data	?	?
booleano	?	?

## Problemas (27/02/2025)

*Estratégia:*

- a) fazer o pseudo-código usando comentários
- b) “recheiar” os comentários com código
- c) os códigos deverão ser feitos em **python** e em **C** (duas versões do mesmo código)
  - i) <https://www.w3schools.com/c>
  - ii) <https://www.w3schools.com/python>
- d) os códigos deverão ser armazenados em repositório de código
  - i) <https://hvescovi.github.io/site/teaching.html>

## 1) Média (27/02/2025)

**Faça um programa que leia 10 números e calcule a média desses números.**

- a) fazer uma versão utilizando cálculo ao final dos números
- b) fazer uma versão utilizando cálculo a cada número novo; pode mostrar resultados parciais
- c) variar as versões a) e b) com grandes quantidades de números (pode usar random() para gerar números em vez de pedir, ou pode usar um número fixo); medir tempo de execução durante a execução do programa
  - i) medir tempo:

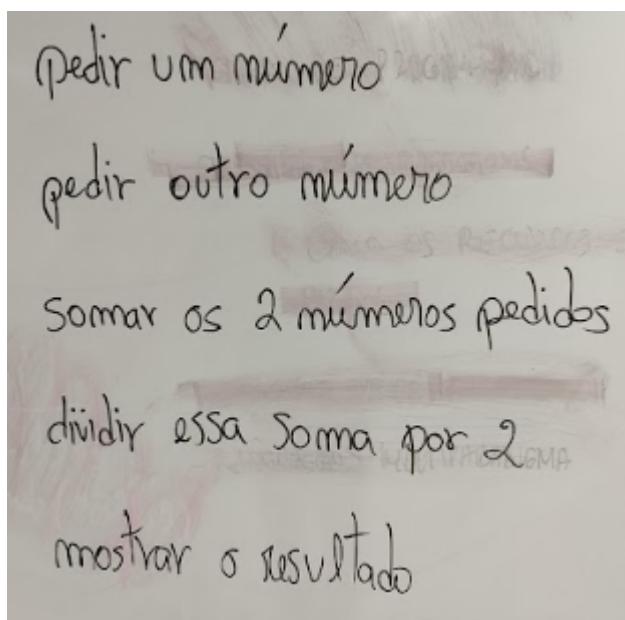
(1) Linux:

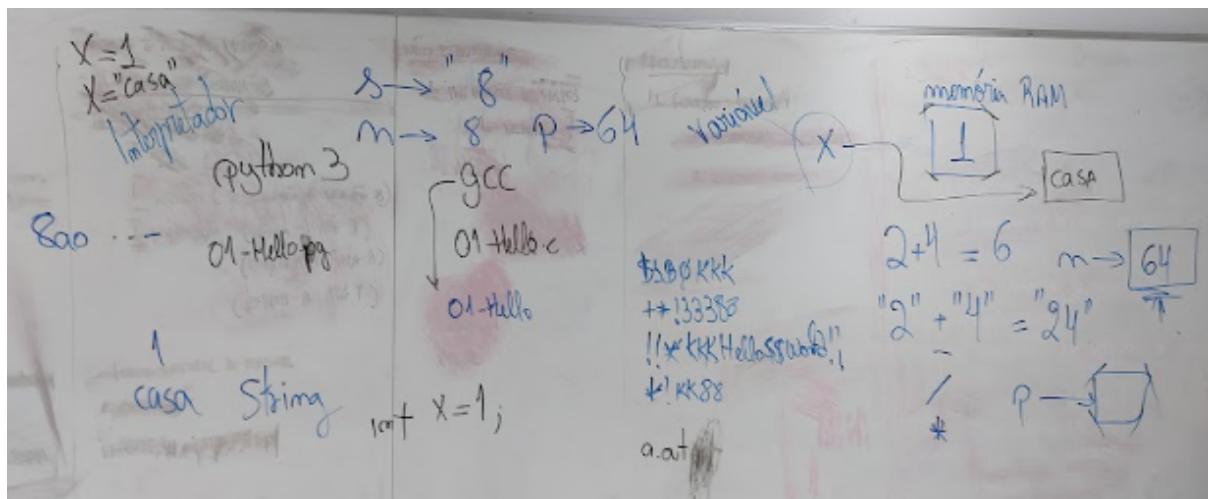
- (a) \$ time python3 hello.py
- (b) \$ /usr/bin/time -v sleep 3

(2) Windows:

- (a) <https://www.pc-tools.net/win32/ptime/>

Explicações do primeiro dia de aula:





## 2) Ordenação (13/03/2025)

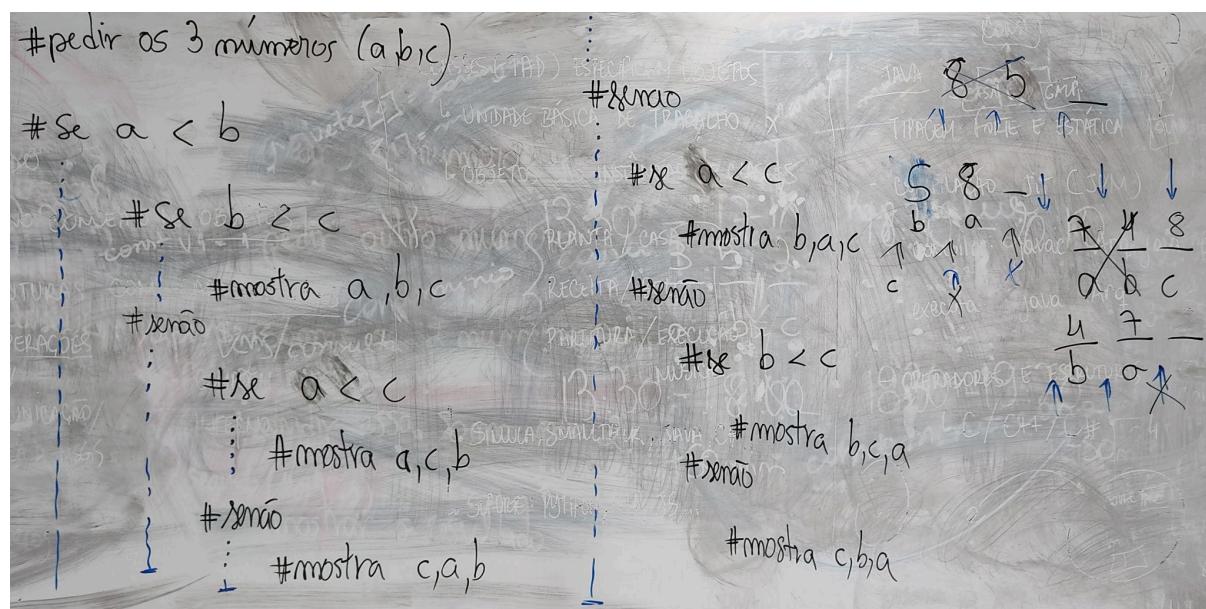
**Faça um programa que leia 10 números e mostre esses números em ordem crescente e em ordem decrescente.**

- se quiser, faça uma versão com 2 números apenas, usando variáveis simples
- se quiser, faça uma versão com 3 números apenas, usando variáveis simples
- para quantidades maiores de números, é recomendado utilizar uma estrutura de dados como um vetor para armazenar os números
  - como ordenar os números?
- faça as variações de quantidade de números e as medições conforme detalhamentos no item c) do problema 1)

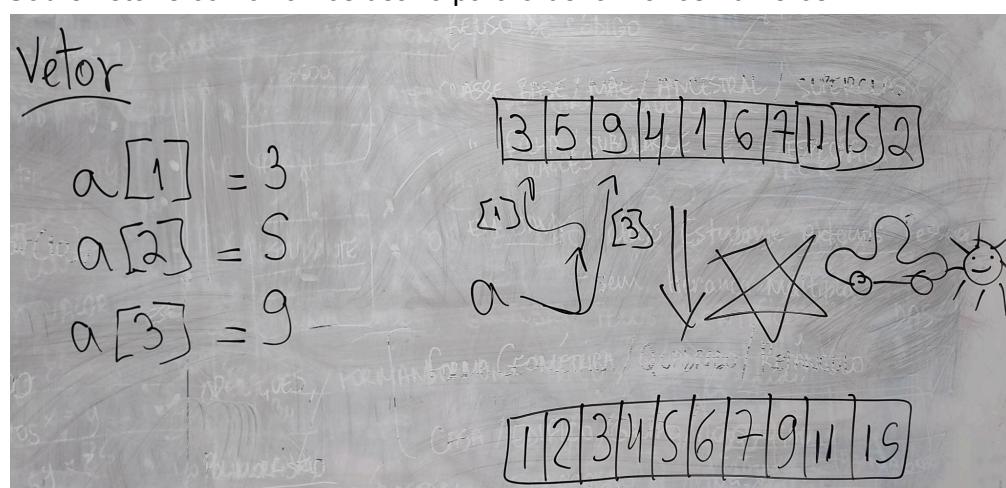
Ordenando apenas 2 números:

```
# Pedir um número (a)
a = int(input("Dig um núm"))
# pedir outro número (b)
b = input("Outro núm.")
# Se a for menor do que b
if a < b:
    #mostre a e depois b
    print(a, b)
# Senão
else:
    #mostre b e depois a
    print(b, a)
```

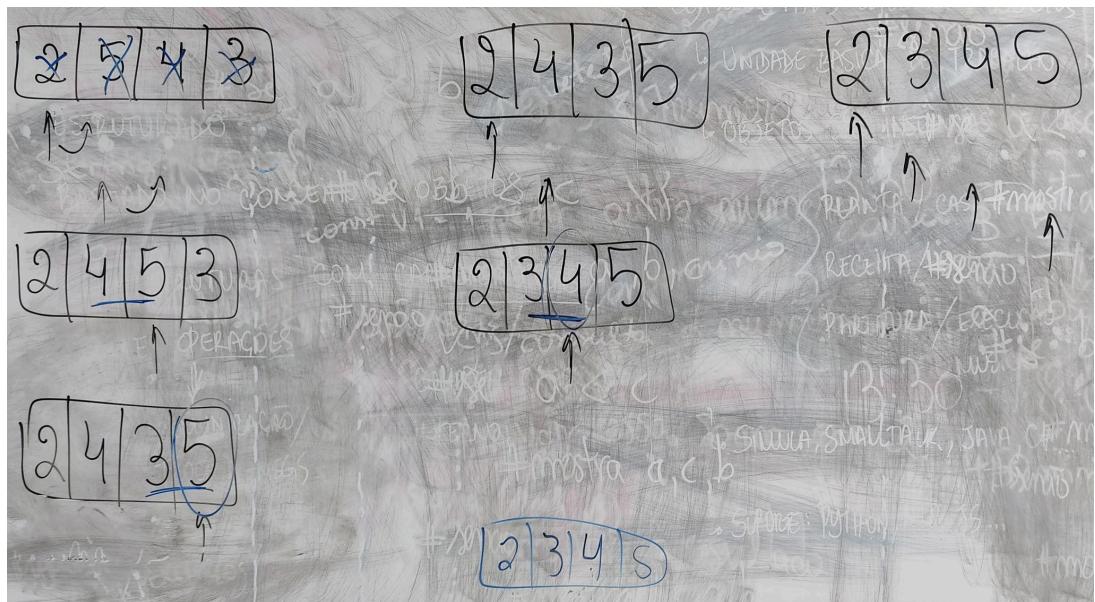
Ordenando 3 números:



Sobre vetor e como vamos usá-lo para ordenar vários números:



Uma estratégia de ordenação de números (“percorrer o vetor fazendo trocas até que não haja mais trocas”)



Algoritmo para ordenar (fazer as trocas):

```
# Comece
  : TIPAGEM: Fazendo bubble sort
  : # Supor que não haver troca
  : 
  : # Percorre o vetor inteiro, menos 1, até o penúltimo
  : 
  :     # Se o elemento atual (i) é maior que o próximo...
  :         # Troca os elementos i e i+1 (maior a)
  :         print(a, b)
  : 
  :         # Simula que teve uma troca
  : 
  : # enquanto houver troca.
```

### 3) Leitura de arquivos (27/03/2025)

- i) Faça um programa que leia um arquivo texto que contém dados sobre incêndios (<https://www.kaggle.com/gustavomodelli/forest-fires-in-brazil#amazon.csv>; também em <https://gitlab.com/hvescovi/prog23/-/tree/main/bd/datasets/fire>) e conte quantos incêndios ocorreram em cada estado, em um determinado ano informado pelo usuário.
- ii) Faça um programa que leia um arquivo texto que contém dados sobre companhias (<https://www.kaggle.com/datasets/peopledatalabssf/free-7-million-company-dataset>; também pode ser obtido com o professor) e encontre qual é o nome da companhia mais antiga cadastrada (4a coluna).

The image shows handwritten Python code on a chalkboard. The code defines a function `mostra_linha(x)` which iterates over elements in `x`, prints them, and then calls `mostra_linha(frase1)` and `mostra_linha(frase2)`. To the right, there is a diagram showing `x` pointing to a box labeled `S`, with an asterisk (\*) above the arrow.

```
def mostra_linha(x):
    for c in x:
        print(c)
    mostra_linha(frase1)
    mostra_linha(frase2)

    * → [S]
```

```

191.52.6.170 :8000
# Criar um acumulador (total = 0) → B python
# abrir arquivo and or
# enquanto houver linhas: (linha) C
# quebrar a linha em partes (partes)
# se amo = 2010 e estado = Santa Catarina
# somar ao acumulador a qtd de incendios (parte 4)
# mostrar o contador

```

A	B	$A \wedge B$	$A \vee B$
amo = 2010	estado = SC		
F	F	F	F
F	V	F	V
V	F	F	V
> V	V	V	V

#### 4) Recursividade (02/04/2025)

- a) Implemente uma função que calcule o fatorial de um número
  - i) Implementar de uma maneira recursiva e de uma maneira não recursiva
  - ii) Organizar cada função em um arquivo diferente
  - iii) Medir tempos de execução em python e em C, versão recursiva e versão procedural
  - iv) Usar valores grandes para ressaltar diferença

- b) Faça um programa que, dado “n” (solicitado ao usuário) e uma sequência com “n” números inteiros (informados pelo usuário), imprima a sequência na ordem inversa a que foi lida, sem usar vetor.

Explicações da aula:

Recursividade

$$(m)! = 5 \times 4 \times 3 \times 2 \times 1 = 1 \times 2 \times 3 \times 4 \times 5$$

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

$$5! = 5 \cdot 4!$$

$$4! = 4 \cdot 3!$$

$$m! = m \cdot (m-1)!$$

$$(m-1)! = (m-1) \cdot (m-2)!$$

$$0! = 1$$

Diagram illustrating the recursive breakdown of factorials. It shows a sequence of numbers from 1 to 120, with arrows indicating the multiplication steps:  $1 \times 2 \times 3 \times 4 \times 5$ ,  $1 \times 2 \times 3 \times 4 \times 6$ ,  $1 \times 2 \times 3 \times 4 \times 120$ . Red annotations highlight the step from  $4! = 4 \cdot 3!$  to  $5! = 5 \cdot 4!$ .

```

# define multiplicador como 1 (m)
# define m (5)
# varie um numero de 1 ate m (i)
# multiplique m * i e armazene em m
# mostre m

```

The pseudocode is annotated with comments explaining the steps:

- # define multiplicador como 1 (m)
- # define m (5)
- # varie um numero de 1 ate m (i)
- # multiplique m \* i e armazene em m
- # mostre m

Below the pseudocode, there is a diagram showing three boxes labeled  $m=1$ ,  $m=2$ , and  $m=3$ , likely representing the state of variable  $m$  at different stages of the loop.

```

# definir função factorial, recebendo m
def fat(m):
    # se m = 0
    if m == 0:
        # retorna 1
        return 1
    # senão
    else:
        # retorna m vezes o factorial de (m-1)
        return m * fat(m-1)

```

3. ~~2~~  
2. ~~1~~  
1. ~~1~~

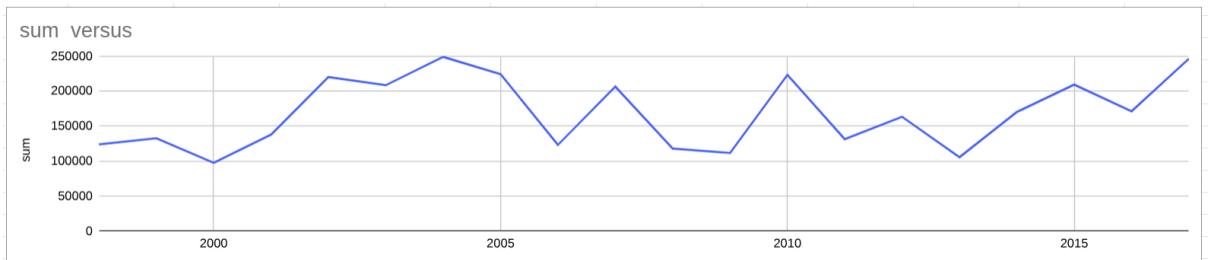
```

# pede factorial de zero (a)
a = fat(0)
# mostra (a)
print(a)
# pede factorial de 3 (b)
b = fat(3)
# mostra b
print(b)

```

## 5) Gráficos (12/06/2025)

- Verifique os programas básicos que geram gráficos em python neste repositório:  
[https://gitlab.com/hvescovi/prog25/-/tree/main/graphics/python/mathplot?ref\\_type=heads](https://gitlab.com/hvescovi/prog25/-/tree/main/graphics/python/mathplot?ref_type=heads)
- Utilizando as bases de dados do item 3) faça um programa que:
  - Some as quantidades de incêndios por ano e gere um gráfico. Um exemplo de gráfico que pode ser gerado é este:



- ii) Mostre quantas companhias foram fundadas por ano em um gráfico.

## Exercícios selecionados do BeeCrowd (02/04/2025)

- Fórmula de Báskara: <https://judge.beecrowd.com/pt/problems/view/1036>
- Média ponderada: <https://judge.beecrowd.com/pt/problems/view/1006>
- Consumo de combustível: <https://judge.beecrowd.com/pt/problems/view/1017>
- Tipos de triângulos: <https://judge.beecrowd.com/pt/problems/view/1045>
- Tabuada: <https://judge.beecrowd.com/pt/problems/view/1078>
- Tipo de combustível: <https://judge.beecrowd.com/pt/problems/view/1134>
- Sequência S: <https://judge.beecrowd.com/pt/problems/view/1155>
- Troca em vetor I: <https://judge.beecrowd.com/pt/problems/view/1175>
- A corrida de lesmas: <https://judge.beecrowd.com/pt/problems/view/1789>

## Exercícios diversos (19/03/2025)

- No programa 07-if, como fazer com que não sejam mostrados/utilizados números repetidos nos sorteios?

## Exercícios preparatórios para a AV1 (23/04/2025)

Sugere-se que os programas a seguir sejam feitos primeiramente com pseudo-código e depois implementados em python ou C.

- Faça um programa que gere cinquenta mil números inteiros positivos diferentes. Esses números devem ser salvos em um arquivo texto, sendo que em cada linha deve haver um número. Além disso, os números devem estar fora de ordem.
  - Sugestão: faça antes um programa para gerar 10 números, depois 100 e depois a quantidade pedida no exercício.
- Faça um programa que ordene o arquivo gerado na questão anterior.
- Faça um programa que ordene o arquivo gerado na questão 1 usando um algoritmo diferente do algoritmo usado na questão 2.
- Compare os tempos de execução dos algoritmos executados nas questões 2 e 3.
- Faça um programa que resolva a seguinte situação: sabe-se que 4 pintores conseguem renovar a pintura de 6 escolas em 8 dias. Considerando que temos 8 pintores que deverão dar conta de 18 escolas, quantos dias serão necessários para

realizar o trabalho?

Referência: <https://brasilescola.uol.com.br/matematica/regra-tres-composta.htm>