



Introdução ao Processamento de Dados

Turma 3 (2020.1)



Ordenação e Busca

Gilson. A. O. P. Costa (IME/UERJ)

gilson.costa@ime.uerj.br

Ordenação

- Ordenação significa colocar um conjuntos de valores em alguma **ordem específica**: crescente, decrescente, alfabética, etc.
- Exemplo:
 notas = [7.5, 8.0, 5.5, 9.0, 8.0, 4.5] # Lista não ordenada.
 notas = [4.5, 5.5, 7.5, 8.0, 8.0, 9.0] # Lista ordenada.
- Por que ordenar?
- A ordenação é fundamental para fazer uma **busca eficiente**.

Ordenação

- A ordenação é fundamental para fazer uma **busca eficiente**.
- É muito mais fácil (e rápido) procurar um valor em uma lista ordenada.
- Imagine que se queira encontrar o nome (ou o CPF) de uma pessoa numa lista de usuários de algum serviço.
- Se esta lista for pequena, não faz muita diferença, mas se a lista contiver **dezenas de milhões de elementos** (como os de uma operadora de telefonia) é fundamental que a busca seja muito rápida.
- Imagine se este conjunto tiver **alguns bilhões de elementos** (existem mais de 4 bilhões de sites na Internet).

Ordenação

- Como ordenar um vetor?

notas = [7.5, 8.0, 5.5, 9.0, 8.0, 4.5]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.

Ordenação

- Como ordenar um vetor?

notas = [7.5, 8.0, 5.5, 9.0, 8.0, 4.5]

Ordenação por seleção (*selection sort*):

- **Procurar o elemento do vetor com o menor valor.**
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.

Ordenação

- Como ordenar um vetor?

notas = [7.5, 8.0, 5.5, 9.0, 8.0, **4.5**]

Ordenação por seleção (*selection sort*):

- **Procurar o elemento do vetor com o menor valor.**
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.

Ordenação

- Como ordenar um vetor?

notas = [7.5, 8.0, 5.5, 9.0, 8.0, **4.5**]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- **Trocar o valor deste elemento com o do primeiro elemento do vetor.**
- Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.

Ordenação

- Como ordenar um vetor?

notas = [4.5, 8.0, 5.5, 9.0, 8.0, 7.5]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- **Trocar o valor deste elemento com o do primeiro elemento do vetor.**
- Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.

Ordenação

- Como ordenar um vetor?

notas = [4.5, 8.0, 5.5, 9.0, 8.0, 7.5]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 8.0, 5.5, 9.0, 8.0, 7.5]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 8.0, 9.0, 8.0, 7.5]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 8.0, 9.0, 8.0, 7.5]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 8.0, 9.0, 8.0, 7.5]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 9.0, 8.0, 8.0]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 9.0, 8.0, 8.0]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 9.0, 8.0, 8.0]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 8.0, 9.0, 8.0]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 8.0, 9.0, 8.0]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 8.0, 9.0, 8.0]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 8.0, 8.0, 9.0]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 8.0, 8.0, 9.0]



Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação

- Como ordenar um vetor?

notas = [4.5, 5.5, 7.5, 8.0, 8.0, 9.0]

Ordenação por seleção (*selection sort*):

- Procurar o elemento do vetor com o menor valor.
- Trocar o valor deste elemento com o do primeiro elemento do vetor.
- **Repetir para o restante do vetor, considerando que o primeiro elemento do vetor já está ordenado.**

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    pos_min = i
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
        if notas[j]<notas[pos_min]:
            pos_min = j
    temp = notas[i]          # troca as notas
    notas[i] = notas[pos_min]
    notas[pos_min] = temp
print(notas)
```

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))  
notas = [0.0]*N  
for i in range(0,N):  
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))  
print(notas)  
for i in range(0,N-1):  
    pos_min = i  
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1  
        if notas[j]<notas[pos_min]:  
            pos_min = j  
    temp = notas[i]          # troca as notas  
    notas[i] = notas[pos_min]  
    notas[pos_min] = temp  
print(notas)
```

[5.5, 4.5, 8.0, 9.0, 7.5]

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=0

[5.5, 4.5, 8.0, 9.0, 7.5]

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=0

[5.5, 4.5, 8.0, 9.0, 7.5]

j=1

pos_min=0

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    pos_min = i
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
        if notas[j]<notas[pos_min]:
            pos_min = j
    temp = notas[i]          # troca as notas
    notas[i] = notas[pos_min]
    notas[pos_min] = temp
print(notas)
```

i=0
[5.5, 4.5, 8.0, 9.0, 7.5]
j=1
pos_min=0

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=0

[5.5, 4.5, 8.0, 9.0, 7.5]

j=1

pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=0

[5.5, 4.5, 8.0, 9.0, 7.5]

j=2

pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=0

[5.5, 4.5, 8.0, 9.0, 7.5]

j=3

pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=0

[5.5, 4.5, 8.0, 9.0, 7.5]

j=4

pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    pos_min = i
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
        if notas[j]<notas[pos_min]:
            pos_min = j
    temp = notas[i]          # troca as notas
    notas[i] = notas[pos_min]
    notas[pos_min] = temp
print(notas)
```

i=0
[5.5, 4.5, 8.0, 9.0, 7.5]
j=4
pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    pos_min = i
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
        if notas[j]<notas[pos_min]:
            pos_min = j
    temp = notas[i]          # troca as notas
    notas[i] = notas[pos_min]
    notas[pos_min] = temp
print(notas)
```

i=0
[4.5, 5.5, 8.0, 9.0, 7.5]
j=4
pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=1

[4.5, 5.5, 8.0, 9.0, 7.5]

j=2

pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=1

[4.5, 5.5, 8.0, 9.0, 7.5]

j=3

pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=1

[4.5, 5.5, 8.0, 9.0, 7.5]

j=4

pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    pos_min = i
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
        if notas[j]<notas[pos_min]:
            pos_min = j
    temp = notas[i]          # troca as notas
    notas[i] = notas[pos_min]
    notas[pos_min] = temp
print(notas)
```

i=1
[4.5, 5.5, 8.0, 9.0, 7.5]
j=4
pos_min=1

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=2

[4.5, 5.5, 8.0, 9.0, 7.5]

j=3

pos_min=2

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=2

[4.5, 5.5, 8.0, 9.0, 7.5]

j=4

pos_min=4

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=2

[4.5, 5.5, 8.0, 9.0, 7.5]

j=4

pos_min=4

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    pos_min = i
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
        if notas[j]<notas[pos_min]:
            pos_min = j
    temp = notas[i]           # troca as notas
    notas[i] = notas[pos_min]
    notas[pos_min] = temp
print(notas)
```

i=2
[4.5, 5.5, 7.5, 9.0, 8.0]
j=4
pos_min=4

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
```

```
notas = [0.0]*N
```

```
for i in range(0,N):
```

```
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
```

```
print(notas)
```

```
for i in range(0,N-1):
```

```
    pos_min = i
```

```
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
```

```
        if notas[j]<notas[pos_min]:
```

```
            pos_min = j
```

```
    temp = notas[i]          # troca as notas
```

```
    notas[i] = notas[pos_min]
```

```
    notas[pos_min] = temp
```

```
print(notas)
```

i=3

[4.5, 5.5, 7.5, 9.0, 8.0]

j=4

pos_min=3

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    pos_min = i
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
        if notas[j]<notas[pos_min]:
            pos_min = j
    temp = notas[i]          # troca as notas
    notas[i] = notas[pos_min]
    notas[pos_min] = temp
print(notas)
```

i=3
[4.5, 5.5, 7.5, 9.0, 8.0]
j=4
pos_min=4

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    pos_min = i
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1
        if notas[j]<notas[pos_min]:
            pos_min = j
    temp = notas[i]           # troca as notas
    notas[i] = notas[pos_min]
    notas[pos_min] = temp
print(notas)
```

i=3
[4.5, 5.5, 7.5, 8.0, 9.0]
j=4
pos_min=4

Ordenação por Seleção

```
N = int(input("Quanto alunos há na turma? "))  
notas = [0.0]*N  
for i in range(0,N):  
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))  
print(notas)  
for i in range(0,N-1):  
    pos_min = i  
    for j in range(i+1,N):    # procura o mínimo entre i+1 e N-1  
        if notas[j]<notas[pos_min]:  
            pos_min = j  
    temp = notas[i]          # troca as notas  
    notas[i] = notas[pos_min]  
    notas[pos_min] = temp  
print(notas)
```

[4.5, 5.5, 7.5, 8.0, 9.0]

Método da Bolha

- Método da bolha (*bubble sort*).

notas = [7.5, 8.0, 5.5, 9.0, 4.5]

- Compara o elemento i com todos os outros elementos $j > i$.
- Se algum elemento (na posição j) tiver valor menor que o da posição i , troca os valores das duas posições.
- Repetir para o próximo valor de i .

Método da Bolha

notas = [7.5, 8.0, 5.5, 9.0, 4.5]

Método da Bolha

notas = [**7.5**, **8.0**, 5.5, 9.0, 4.5] # $i=0$; $j=1$

notas = [**7.5**, 8.0, **5.5**, 9.0, 4.5] # $i=0$; $j=2$

Método da Bolha

notas = [**7.5**, **8.0**, 5.5, 9.0, 4.5] # $i=0$; $j=1$

notas = [**5.5**, 8.0, **7.5**, 9.0, 4.5] # $i=0$; $j=2$

notas = [**5.5**, 8.0, 7.5, **9.0**, 4.5] # $i=0$; $j=3$

notas = [**5.5**, 8.0, 7.5, 9.0, **4.5**] # $i=0$; $j=4$

Método da Bolha

notas = [7.5, 8.0, 5.5, 9.0, 4.5] # $i=0; j=1$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0; j=2$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0; j=3$

notas = [4.5, 8.0, 7.5, 9.0, 5.5] # $i=0; j=4$

notas = [4.5, 8.0, 7.5, 9.0, 5.5] # $i=1; j=2$

Método da Bolha

notas = [7.5, 8.0, 5.5, 9.0, 4.5] # $i=0$; $j=1$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0$; $j=2$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0$; $j=3$

notas = [4.5, 8.0, 7.5, 9.0, 5.5] # $i=0$; $j=4$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1$; $j=2$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1$; $j=3$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1$; $j=4$

Método da Bolha

notas = [7.5, 8.0, 5.5, 9.0, 4.5] # $i=0; j=1$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0; j=2$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0; j=3$

notas = [4.5, 8.0, 7.5, 9.0, 5.5] # $i=0; j=4$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1; j=2$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1; j=3$

notas = [4.5, 5.5, 8.0, 9.0, 7.5] # $i=1; j=4$

notas = [4.5, 5.5, 8.0, 9.0, 7.5] # $i=2; j=3$

notas = [4.5, 5.5, 8.0, 9.0, 7.5] # $i=2; j=4$

Método da Bolha

notas = [7.5, 8.0, 5.5, 9.0, 4.5] # $i=0; j=1$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0; j=2$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0; j=3$

notas = [4.5, 8.0, 7.5, 9.0, 5.5] # $i=0; j=4$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1; j=2$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1; j=3$

notas = [4.5, 5.5, 8.0, 9.0, 7.5] # $i=1; j=4$

notas = [4.5, 5.5, 8.0, 9.0, 7.5] # $i=2; j=3$

notas = [4.5, 5.5, 7.5, 9.0, 8.0] # $i=2; j=4$

notas = [4.5, 5.5, 7.5, 9.0, 8.0] # $i=3; j=4$

Método da Bolha

notas = [7.5, 8.0, 5.5, 9.0, 4.5] # $i=0; j=1$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0; j=2$

notas = [5.5, 8.0, 7.5, 9.0, 4.5] # $i=0; j=3$

notas = [4.5, 8.0, 7.5, 9.0, 5.5] # $i=0; j=4$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1; j=2$

notas = [4.5, 7.5, 8.0, 9.0, 5.5] # $i=1; j=3$

notas = [4.5, 5.5, 8.0, 9.0, 7.5] # $i=1; j=4$

notas = [4.5, 5.5, 8.0, 9.0, 7.5] # $i=2; j=3$

notas = [4.5, 5.5, 7.5, 9.0, 8.0] # $i=2; j=4$

notas = [4.5, 5.5, 7.5, 8.0, 9.0] # $i=3; j=4$

notas = [4.5, 5.5, 7.5, 8.0, 9.0]

Método da Bolha

```
N = int(input("Quanto alunos há na turma? "))
notas = [0.0]*N
for i in range(0,N):
    notas[i] = float(input("Entre com a nota #{}: ".format(i+1)))
print(notas)
for i in range(0,N-1):
    for j in range(i+1,N):
        if notas[i] > notas[j]:
            aux = notas[i]
            notas[i] = notas[j]
            notas[j] = aux
print(notas)
```

Ordenação

- O método da classe lista ***sort()***, ordena a lista/vetor.

```
>>> vet = [3, 2, 4, 1]
```

```
>>> vet.sort()
```

```
>>> print(vet)
```

```
>>> [1, 2, 3, 4]
```

- Neste curso, vocês estão proibidos de usar o método ***sort()***!
- A ideia é que exercícios de ordenação ajudam a fixar características importantes das listas/vetores.

Ordenação

- O método da classe lista ***sort()***, ordena a lista/vetor.

```
>>> vet = [3, 2, 4, 1]
```

```
>>> vet.sort()
```

```
>>> print(vet)
```

```
>>> [1, 2, 3, 4]
```

- Neste curso, vocês estão proibidos de usar o método ***sort()***!
- A ideia é que exercícios de ordenação ajudam a fixar características importantes das listas/vetores.

Busca

- Objetivo é **procurar um valor específico** num vetor/lista.
- Se o vetor não estiver ordenado, única alternativa é a **busca exaustiva**.
- Avalia cada elemento do vetor até encontrar o valor procurado.

Busca Exaustiva

```
vetor = [7.5, 8.0, 5.5, 9.0, 4.5, 8.0, 5.5, 6.0, 10.0]
valor = float(input('Entre com o valor a procurar: '))
posicao = -1
for i in range(0, len(vetor)):
    if vetor[i] == valor:
        posicao = i
if posicao >= 0:
    print('Valor encontrado na posição {}'.format(posicao))
else:
    print('Valor não encontrado.')
```

Busca Exaustiva

Parando a busca quando encontra a primeira ocorrência do valor:

```
vetor = [7.5, 8.0, 5.5, 9.0, 4.5, 8.0, 5.5, 6.0, 10.0]
valor = float(input('Entre com o valor a procurar: '))
posicao = 0
achou = False
while (not achou) and (posicao < len(vetor)):
    if vetor[posicao] == valor:
        achou = True
    else:
        posicao = posicao + 1
if achou:
    print('Valor encontrado na posição {}'.format(posicao))
else:
    print('Valor não encontrado.')
```

Busca Exaustiva

Indicando o número de vezes e posições em que o valor foi encontrado:

```
vetor = [7.5, 8.0, 5.5, 9.0, 4.5, 8.0, 5.5, 6.0, 10.0]
valor = float(input('Entre com o valor a procurar: '))
posicoes = []
for i in range(0, len(vetor)):
    if vetor[i] == valor:
        posicoes.append(i)
if len(posicoes) == 1:
    print('Valor encontrado na posição: ', posicoes)
elif len(posicoes) != 0:
    print('Valor encontrado nas posições: ', posicoes)
else:
    print('Valor não encontrado.')
```

Busca Binária

- Se o vetor/lista estiver ordenado pode-se fazer uma busca muito mais eficiente: **busca binária**.
- A cada iteração, o problema é **reduzido à metade**.
- Compara-se o valor procurado com o do elemento do meio do vetor.
- Caso os valores sejam iguais: termina a busca.
- Caso contrário:
 - se o **valor do elemento do meio for maior** que o valor procurado, fazer a **busca apenas na metade inferior do vetor**.
 - se o **valor do elemento do meio for menor** que o valor procurado, fazer a **busca apenas na metade superior do vetor**.

Busca Binária

`vetor = [4.5, 5.0, 5.5, 6.0, 7.5, 8.0, 8.5, 9.0, 10.0]`

`inicio=0`

`valor_procurado=6.0`

`fim=8`

`meio=(inicio+fim)//2=4`

Busca Binária

vetor = [4.5, 5.0, 5.5, 6.0, 7.5, 8.0, 8.5, 9.0, 10.0]

inicio=0

valor_procurado=6.0

fim=8

meio=(inicio+fim)//2=4

vetor[meio]=7.5

Busca Binária

vetor = [4.5, 5.0, 5.5, 6.0, 7.5, 8.0, 8.5, 9.0, 10.0]

início=0

valor_procurado=6.0

fim=meio-1=3

Busca Binária

vetor = [4.5, **5.0**, 5.5, 6.0, 7.5, 8.0, 8.5, 9.0, 10.0]

inicio=0

valor_procurado=6.0

fim=meio-1=3

meio=(inicio+fim)//2=1

vetor[meio]=5.0

Busca Binária

vetor = [4.5, 5.0, 5.5, 6.0, 7.5, 8.0, 8.5, 9.0, 10.0]

início=meio+1=2

valor_procurado=6.0

fim=3

meio=(início+fim)//2=2

Busca Binária

vetor = [4.5, 5.0, **5.5, 6.0**, 7.5, 8.0, 8.5, 9.0, 10.0]

início=meio+1=2

valor_procurado=6.0

fim=3

meio=(início+fim)//2=2

vetor[meio]=5.5

Busca Binária

vetor = [4.5, 5.0, 5.5, 6.0, 7.5, 8.0, 8.5, 9.0, 10.0]

início=meio+1=3

valor_procurado=6.0

fim=3

meio=(início+fim)//2=3

vetor[meio]=6.0

Busca Binária

`vetor = [4.5, 5.0, 5.5, 6.0, 7.5, 8.0, 8.5, 9.0, 10.0]`

`inicio=meio+1=3`

`valor_procurado=6.0`

`fim=3`

`meio=(inicio+fim)//2=3`

`vetor[meio]=6.0`

Busca Binária

```
vetor = [4.5, 5.0, 5.5, 6.0, 7.5, 8.0, 8.5, 9.0, 10.0]
valor = float(input('Entre com o valor a procurar: '))
inicio = 0
fim = len(vetor)-1
achou = False
while (not achou) and (inicio <= fim):
    meio = (inicio + fim)//2
    if valor == vetor[meio]:
        achou = True
    elif valor < vetor[meio]:
        fim = meio-1
    else:
        inicio = meio+1
if achou:
    print('Valor encontrado na posição: ', meio)
else:
    print('Valor não encontrado.')
```

Estatísticas

- Já tínhamos as ferramentas para calcular a média, máximo, mínimo e o desvio padrão de um conjunto de valores.
- Agora podemos calcular outra estatística importante: a **mediana** destes valores.

[3.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.5, 8.0, 8.0, 9.0, 9.0, 9.0, 10.0]

Estatísticas

- Já tínhamos as ferramentas para calcular a média, máximo, mínimo e o desvio padrão de um conjunto de valores.
- Agora podemos calcular outra estatística importante: a **mediana** destes valores.

[3.0, 4.5, 5.0, 5.5, 6.0, 6.5, **7.5**, 8.0, 8.0, 9.0, 9.0, 9.0, 10.0]

- A mediana é o **valor central do conjunto** (ordenado).
- Se o número de elementos for par, a mediana é a média dos valores centrais.

Estatísticas

- Outra estatística importante é a **moda**.
- A moda indica o valor que aparece com maior frequência.

[3.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.5, 8.0, 8.0, 9.0, 9.0, 9.0, 10.0]

Estatísticas

- Outra estatística importante é a **moda**.
- A moda indica o valor que aparece com maior frequência.

[3.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.5, 8.0, 8.0, **9.0, 9.0, 9.0**, 10.0]

- Primeira moda = 9.0
- Segunda moda = 8.0

Exercício

Calcule as seguintes estatísticas para o conjunto de notas de uma determinada turma: máximo, mínimo, média, desvio padrão, mediana e moda (primeira).

```
turma1 = [1.1, 7.5, 0.8, 1.8, 1.5, 1.9, 10.0, 10.0, 9.3, 10.0, 7.7, 0.6, 0.5, 8.7,  
5.6, 7.0, 8.3, 7.0, 9.1, 7.4, 8.1, 7.0, 6.3, 0.6, 7.4, 2.8, 5.0, 1.4, 1.5, 0.5, 8.3,  
7.0, 2.9, 7.6, 10.0, 3.3, 1.9, 5.1, 7.0]
```

```
turma2 = [10.0, 8.2, 8.7, 5.5, 6.8, 8.6, 8.5, 6.1, 6.2, 8.5, 7.7, 10.0, 10.0, 6.1,  
8.4, 5.4, 5.6, 9.8, 2.1, 8.5, 3.3, 8.7, 8.5, 9.1, 9.7]
```



Introdução ao Processamento de Dados

Turma 3 (2020.1)



Ordenação e Busca

Gilson. A. O. P. Costa (IME/UERJ)

gilson.costa@ime.uerj.br