

Tipos sequenciais

Aula 10

Diego Padilha Rubert

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Algoritmos e Programação

- 1 Motivação
- 2 Lista
- 3 Exemplo com listas
- 4 Exercícios

- ▶ tipos numéricos de dados se caracterizam pelo fato que apenas um valor pode ser armazenado em uma variável
- ▶ tipos sequenciais armazenam uma sequência de valores
- ▶ já conhecemos um tipo sequencial que armazena uma sequência de letras: o tipo *string*
- ▶ existem vários outros tipos de dados compostos que não veremos aqui, como a tupla (*tuple*), o dicionário (*dict*) e o conjunto (*set*), etc, mas alguns desses tipos não necessariamente armazenam elementos de forma sequencial

- ▶ tipos numéricos de dados se caracterizam pelo fato que apenas um valor pode ser armazenado em uma variável
- ▶ tipos sequenciais armazenam uma sequência de valores
- ▶ já conhecemos um tipo sequencial que armazena uma sequência de letras: o tipo *string*
- ▶ existem vários outros tipos de dados compostos que não veremos aqui, como a tupla (*tuple*), o dicionário (*dict*) e o conjunto (*set*), etc, mas alguns desses tipos não necessariamente armazenam elementos de forma sequencial

- ▶ tipos numéricos de dados se caracterizam pelo fato que apenas um valor pode ser armazenado em uma variável
- ▶ tipos sequenciais armazenam uma sequência de valores
- ▶ já conhecemos um tipo sequencial que armazena uma sequência de letras: o tipo *string*
- ▶ existem vários outros tipos de dados compostos que não veremos aqui, como a tupla (*tuple*), o dicionário (*dict*) e o conjunto (*set*), etc, mas alguns desses tipos não necessariamente armazenam elementos de forma sequencial

- ▶ tipos numéricos de dados se caracterizam pelo fato que apenas um valor pode ser armazenado em uma variável
- ▶ tipos sequenciais armazenam uma sequência de valores
- ▶ já conhecemos um tipo sequencial que armazena uma sequência de letras: o tipo *string*
- ▶ existem vários outros tipos de dados compostos que não veremos aqui, como a tupla (*tuple*), o dicionário (*dict*) e o conjunto (*set*), etc, mas alguns desses tipos não necessariamente armazenam elementos de forma sequencial

Problema: dadas cinco notas de uma prova dos(as) estudantes de uma disciplina, calcular a média das notas da prova e a quantidade de estudantes que obtiveram nota maior que a média e a quantidade de estudantes que obtiveram nota menor que a média

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

nota1,nota2,nota3,nota4,nota5 = input("Informe as notas: ").split()
nota1,nota2,nota3 = float(nota1),float(nota2),float(nota3)
nota4,nota5 = float(nota4),float(nota5)
media = (nota1 + nota2 + nota3 + nota4 + nota5) / 5
print("Média das provas: %f" % media)
```

Problema: dadas cinco notas de uma prova dos(as) estudantes de uma disciplina, calcular a média das notas da prova e a quantidade de estudantes que obtiveram nota maior que a média e a quantidade de estudantes que obtiveram nota menor que a média

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

nota1,nota2,nota3,nota4,nota5 = input("Informe as notas: ").split()
nota1,nota2,nota3 = float(nota1),float(nota2),float(nota3)
nota4,nota5 = float(nota4),float(nota5)
media = (nota1 + nota2 + nota3 + nota4 + nota5) / 5
print("Média das provas: %f" % media)
```



```
menor = 0
if nota1 < media:
    menor += 1
if nota2 < media:
    menor += 1
if nota3 < media:
    menor += 1
if nota4 < media:
    menor += 1
if nota5 < media:
    menor += 1
```

```
maior = 0
if nota1 > media:
    maior += 1
if nota2 > media:
    maior += 1
if nota3 > media:
    maior += 1
if nota4 > media:
    maior += 1
if nota5 > media:
    maior += 1

print("Quantidade com nota inferior à média: %d" % menor)
print("Quantidade com nota superior à média: %d" % maior)

exit(0)
```

- ▶ o que aconteceria se a sala de aula tivesse mais estudantes, como por exemplo 100? ou 1000 estudantes?
- ▶ as estruturas sequencial e condicional não seriam apropriadas para resolver esse problema, já que o programa teria centenas ou milhares de linhas repetitivas, incorrendo inclusive na possibilidade de propagação de erros e na dificuldade de encontrá-los

- ▶ o que aconteceria se a sala de aula tivesse mais estudantes, como por exemplo 100? ou 1000 estudantes?
- ▶ as estruturas sequencial e condicional não seriam apropriadas para resolver esse problema, já que o programa teria centenas ou milhares de linhas repetitivas, incorrendo inclusive na possibilidade de propagação de erros e na dificuldade de encontrá-los

- ▶ o que aconteceria se a sala de aula tivesse mais estudantes, como por exemplo 100? ou 1000 estudantes?
- ▶ as estruturas sequencial e condicional não seriam apropriadas para resolver esse problema, já que o programa teria centenas ou milhares de linhas repetitivas, incorrendo inclusive na possibilidade de propagação de erros e na dificuldade de encontrá-los

- ▶ uma **lista** (que é similar ao **vetor** de outras linguagens), é uma estrutura de armazenamento de dados que se dispõe de forma linear na memória e é usada para armazenar valores diversos
- ▶ embora cada um dos valores possa ser de um tipo diferente, geralmente armazenamos valores de um mesmo tipo
- ▶ cada célula da lista tem um **endereço** ou **índice** através do qual podemos referenciá-la
- ▶ para uma lista com n itens, os índices vão de 0 a $n - 1$ (assim como nas strings!)

- ▶ uma **lista** (que é similar ao **vetor** de outras linguagens), é uma estrutura de armazenamento de dados que se dispõe de forma linear na memória e é usada para armazenar valores diversos
- ▶ embora cada um dos valores possa ser de um tipo diferente, geralmente armazenamos valores de um mesmo tipo
- ▶ cada célula da lista tem um **endereço** ou **índice** através do qual podemos referenciá-la
- ▶ para uma lista com n itens, os índices vão de 0 a $n - 1$ (assim como nas strings!)

- ▶ uma **lista** (que é similar ao **vetor** de outras linguagens), é uma estrutura de armazenamento de dados que se dispõe de forma linear na memória e é usada para armazenar valores diversos
- ▶ embora cada um dos valores possa ser de um tipo diferente, geralmente armazenamos valores de um mesmo tipo
- ▶ cada célula da lista tem um **endereço** ou **índice** através do qual podemos referenciá-la
- ▶ para uma lista com n itens, os índices vão de 0 a $n - 1$ (assim como nas strings!)

- ▶ uma **lista** (que é similar ao **vetor** de outras linguagens), é uma estrutura de armazenamento de dados que se dispõe de forma linear na memória e é usada para armazenar valores diversos
- ▶ embora cada um dos valores possa ser de um tipo diferente, geralmente armazenamos valores de um mesmo tipo
- ▶ cada célula da lista tem um **endereço** ou **índice** através do qual podemos referenciá-la
- ▶ para uma lista com n itens, os índices vão de 0 a $n - 1$ (assim como nas strings!)

- ▶ podemos criar uma lista vazia e depois adicionar itens

```
lista = []
```

- ▶ podemos criar uma lista com alguns elementos

```
lista = [1, 7, 4]
```

- ▶ podemos criar uma lista a partir de outros conjuntos

```
lista = list(range(2,7))
```

- ▶ podemos criar uma lista vazia e depois adicionar itens

```
lista = []
```

- ▶ podemos criar uma lista com alguns elementos

```
lista = [1, 7, 4]
```

- ▶ podemos criar uma lista a partir de outros conjuntos

```
lista = list(range(2,7))
```

- ▶ podemos criar uma lista vazia e depois adicionar itens

```
lista = []
```

- ▶ podemos criar uma lista com alguns elementos

```
lista = [1, 7, 4]
```

- ▶ podemos criar uma lista a partir de outros conjuntos

```
lista = list(range(2,7))
```

Operações

Algumas operações suportadas por listas, supondo listas **s** e **t**, inteiros **i**, **j** e **k**, e algum valor/variável de qualquer tipo **x**:

Operação	Resultado
<code>x in s</code>	True se algum item de <code>s</code> é igual a <code>x</code> , False c.c
<code>x not in s</code>	True se nenhum item de <code>s</code> é igual a <code>x</code> , False c.c
<code>s + t</code>	devolve a concatenação de <code>s</code> e <code>t</code>
<code>s[i]</code>	<code>i</code> -ésimo item de <code>s</code> (começando em 0)
<code>s[i:j]</code>	fatia de <code>i</code> até <code>j-1</code>
<code>len(s)</code>	devolve o comprimento de <code>s</code>
<code>min(s)</code>	devolve o menor elemento de <code>s</code>
<code>max(s)</code>	devolve o maior elemento de <code>s</code>
<code>sum(s)</code>	devolve a soma dos elementos de <code>s</code>
<code>s.count(x)</code>	devolve o número de ocorrências de <code>x</code> em <code>s</code>
<code>s[i] = x</code>	altera o valor na posição <code>i</code> para <code>x</code>

Operações

Algumas operações suportadas por listas, supondo listas **s** e **t**, inteiros **i**, **j** e **k**, e algum valor/variável de qualquer tipo **x**:

Operação

`del s[slice]`

`s.append(x)`

`s.clear()`

`s.copy()`

`s.extend(t)` ou

`s += t`

`s.insert(i, x)`

`s.remove(x)`

Resultado

remove os elementos do *slice* de *s* (pode ser 1 única posição ou um intervalo)

adiciona *x* ao final da lista *s*

remove todos elementos de *s*

cria uma cópia de *s*

estende *s* com os elementos de *t*

insere *x* em *s* na posição *i* (empurra os elementos posteriores)

remove a primeira ocorrência do valor *x* na lista

Exemplo com listas

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

n = int(input("Digite a quantidade de notas: "))

notas = []
for i in range(n):
    notas.append(float(input()))

media = 0.0
for nota in notas:
    media += nota
media /= n

menor = maior = 0
for nota in notas:
    if nota < media:
        menor += 1
    if nota >= media:
        maior += 1

print("Média: %2.2f, %d menores e %d maiores", media, menor, maior)
exit(0)
```



Exercícios

1. Dado um n inteiro e uma sequência de n números inteiros (um por linha), imprima a sequência em ordem inversa à de leitura.
2. Dado um inteiro n que representa uma quantidade de dias e uma sequência de n valores (um por linha) representando as temperaturas desses dias, responda quais dias tiveram temperatura acima da média.
3. Uma prova consta de 6 questões, cada uma com cinco alternativas identificadas pelas letras A, B, C, D e E. Dado o cartão gabarito da prova (em uma linha) e o cartão de respostas de n estudantes (um por linha), computar o número de acertos de cada um dos estudantes.

Exemplo de entrada:

```
B A C D A E
3
A A C D B E
A A A A A A
B A C D B E
```

Exemplo de saída:

```
4
2
5
```


Exercícios

1. Dado um n inteiro e uma sequência de n números inteiros (um por linha), imprima a sequência em ordem inversa à de leitura.
2. Dado um inteiro n que representa uma quantidade de dias e uma sequência de n valores (um por linha) representando as temperaturas desses dias, responda quais dias tiveram temperatura acima da média.
3. Uma prova consta de 6 questões, cada uma com cinco alternativas identificadas pelas letras A, B, C, D e E. Dado o cartão gabarito da prova (em uma linha) e o cartão de respostas de n estudantes (um por linha), computar o número de acertos de cada um dos estudantes.

Exemplo de entrada:

```
B A C D A E
3
A A C D B E
A A A A A A
B A C D B E
```

Exemplo de saída:

```
4
2
5
```

Exercícios

1. Dado um n inteiro e uma sequência de n números inteiros (um por linha), imprima a sequência em ordem inversa à de leitura.
2. Dado um inteiro n que representa uma quantidade de dias e uma sequência de n valores (um por linha) representando as temperaturas desses dias, responda quais dias tiveram temperatura acima da média.
3. Uma prova consta de 6 questões, cada uma com cinco alternativas identificadas pelas letras A, B, C, D e E. Dado o cartão gabarito da prova (em uma linha) e o cartão de respostas de n estudantes (um por linha), computar o número de acertos de cada um dos estudantes.

Exemplo de entrada:

```
B A C D A E
3
A A C D B E
A A A A A A
B A C D B E
```

Exemplo de saída:

```
4
2
5
```

4. Tentando descobrir se um dado era viciado, um dono de cassino o lançou n vezes. Dados os n resultados dos lançamentos (um por linha), determinar o número de ocorrências de cada face.
Exemplo: se $n = 8$ e os resultados dos lançamentos são 6, 5, 3, 1, 6, 1, 2, 4, então a saída deve ser 2, 1, 1, 1, 1, 2.
5. Um jogador interessado em cassinos deseja fazer um levantamento estatístico simples sobre uma roleta. Para isso, ele fez n lançamentos nesta roleta. Sabendo que uma roleta contém 37 números (de 0 a 36), calcular a frequência de cada número desta roleta nos n lançamentos realizados.
Exemplo: se $n = 10$ e os resultados dos lançamentos são 32, 12, 6, 0, 9, 12, 11, 7, 28, 21, então a saída deve ser 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0.

4. Tentando descobrir se um dado era viciado, um dono de cassino o lançou n vezes. Dados os n resultados dos lançamentos (um por linha), determinar o número de ocorrências de cada face.
Exemplo: se $n = 8$ e os resultados dos lançamentos são 6, 5, 3, 1, 6, 1, 2, 4, então a saída deve ser 2, 1, 1, 1, 1, 2.
5. Um jogador interessado em cassinos deseja fazer um levantamento estatístico simples sobre uma roleta. Para isso, ele fez n lançamentos nesta roleta. Sabendo que uma roleta contém 37 números (de 0 a 36), calcular a frequência de cada número desta roleta nos n lançamentos realizados.
Exemplo: se $n = 10$ e os resultados dos lançamentos são 32, 12, 6, 0, 9, 12, 11, 7, 28, 21, então a saída deve ser 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0.

Exercícios

6. Calcule o valor do polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$ em k pontos distintos. São dados os valores de n (grau do polinômio), com $1 \leq n \leq 100$, de a_0, a_1, \dots, a_n (coeficientes reais do polinômio), de k e dos pontos x_1, x_2, \dots, x_k .
7. Dada uma sequência de números inteiros em uma única linha, encontre e escreva o menor valor (NÃO utilize a função `min`).
8. Dada uma sequência de números inteiros em uma única linha, imprimi-la em ordem crescente de seus valores em uma única linha (NÃO utilize a função `min` ou a função `sort`).
9. Dada uma sequência de números inteiros em uma única linha, escreva essa sequência sem repetições em uma única linha.
10. Dada uma sequência de números inteiros em uma única linha, escreva os elementos repetidos em uma única linha.

Exercícios

6. Calcule o valor do polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$ em k pontos distintos. São dados os valores de n (grau do polinômio), com $1 \leq n \leq 100$, de a_0, a_1, \dots, a_n (coeficientes reais do polinômio), de k e dos pontos x_1, x_2, \dots, x_k .
7. Dada uma sequência de números inteiros em uma única linha, encontre e escreva o menor valor (**NÃO utilize a função `min`**).
8. Dada uma sequência de números inteiros em uma única linha, imprimi-la em ordem crescente de seus valores em uma única linha (**NÃO utilize a função `min` ou a função `sort`**).
9. Dada uma sequência de números inteiros em uma única linha, escreva essa sequência sem repetições em uma única linha.
10. Dada uma sequência de números inteiros em uma única linha, escreva os elementos repetidos em uma única linha.

Exercícios

6. Calcule o valor do polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$ em k pontos distintos. São dados os valores de n (grau do polinômio), com $1 \leq n \leq 100$, de a_0, a_1, \dots, a_n (coeficientes reais do polinômio), de k e dos pontos x_1, x_2, \dots, x_k .
7. Dada uma sequência de números inteiros em uma única linha, encontre e escreva o menor valor (**NÃO utilize a função `min`**).
8. Dada uma sequência de números inteiros em uma única linha, imprimi-la em ordem crescente de seus valores em uma única linha (**NÃO utilize a função `min` ou a função `sort`**).
9. Dada uma sequência de números inteiros em uma única linha, escreva essa sequência sem repetições em uma única linha.
10. Dada uma sequência de números inteiros em uma única linha, escreva os elementos repetidos em uma única linha.

Exercícios

6. Calcule o valor do polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$ em k pontos distintos. São dados os valores de n (grau do polinômio), com $1 \leq n \leq 100$, de a_0, a_1, \dots, a_n (coeficientes reais do polinômio), de k e dos pontos x_1, x_2, \dots, x_k .
7. Dada uma sequência de números inteiros em uma única linha, encontre e escreva o menor valor (**NÃO utilize a função `min`**).
8. Dada uma sequência de números inteiros em uma única linha, imprimi-la em ordem crescente de seus valores em uma única linha (**NÃO utilize a função `min` ou a função `sort`**).
9. Dada uma sequência de números inteiros em uma única linha, escreva essa sequência sem repetições em uma única linha.
10. Dada uma sequência de números inteiros em uma única linha, escreva os elementos repetidos em uma única linha.

6. Calcule o valor do polinômio $p(x) = a_0 + a_1x + \dots + a_nx^n$ em k pontos distintos. São dados os valores de n (grau do polinômio), com $1 \leq n \leq 100$, de a_0, a_1, \dots, a_n (coeficientes reais do polinômio), de k e dos pontos x_1, x_2, \dots, x_k .
7. Dada uma sequência de números inteiros em uma única linha, encontre e escreva o menor valor (**NÃO utilize a função `min`**).
8. Dada uma sequência de números inteiros em uma única linha, imprimi-la em ordem crescente de seus valores em uma única linha (**NÃO utilize a função `min` ou a função `sort`**).
9. Dada uma sequência de números inteiros em uma única linha, escreva essa sequência sem repetições em uma única linha.
10. Dada uma sequência de números inteiros em uma única linha, escreva os elementos repetidos em uma única linha.

11. Dizemos que uma sequência de n elementos, com n par, é **balanceada** se as seguintes somas são todas iguais:
- a soma do maior elemento com o menor elemento;
 - a soma do segundo maior elemento com o segundo menor elemento;
 - a soma do terceiro maior elemento com o terceiro menor elemento;
 - e assim por diante ...

Exemplo:

2 12 3 6 16 15 é uma sequência balanceada,
pois $16 + 2 = 15 + 3 = 12 + 6$.

Dada uma sequência de números inteiros em uma única linha, verificar se essa sequência é balanceada. Suponha que a quantidade de números digitados sempre será **par**.

12. Dado uma sequência de números inteiros em uma única linha, determinar a quantidade de vezes que cada um deles ocorre na própria sequência.

Sequência: -1 3 0 5 0 5 -1 2 -5 0

Saída:

-1 ocorre 2 vez(es)

3 ocorre 1 vez(es)

0 ocorre 3 vez(es)

5 ocorre 2 vez(es)

2 ocorre 1 vez(es)

-5 ocorre 1 vez(es)

Você não deve escrever a saída mais de uma vez para o mesmo valor. Dica: utilize uma lista auxiliar para armazenar para quais valores você já escreveu a saída.