

Algoritmos e Estruturas de Dados

Aula 1

Prof Dr Tanilson Dias dos Santos

Universidade Aberta do Brasil – UAB
Universidade Federal do Tocantins - UFT

Quem sou eu?



- Graduação em Ciência da Computação - UFT;
- Mestrado em Sistemas e Computação - IME;
- Doutorado em Engenharia de Sistemas e Computação - COPPE/PESC - UFRJ;
- Currículo Lattes:
<http://lattes.cnpq.br/2812244463844431>
- Contato: tanilson.dias@uft.edu.br

Quem são vocês?



- Vamos Criar uma mini-bio no Canva;
- **Postar no Fórum da Aula 1:**
- Coloquem uma foto;
- Nome;
- Idade;
- Cidade;
- Uma curiosidade sobre você;
- O que você espera aprender nesta disciplina.

Plano de Ensino

PLANO DE ENSINO



Plano de Curso - Algoritmos e Estruturas de Dados

- **Ementa:** Análise de algoritmos. Estruturas de dados lineares: pilhas, filas e listas encadeadas. Estrutura de dados não lineares: árvores. Fila de prioridade e Heaps. Dicionário de dados. Classificação de dados. Balanceamento em árvores.

Dinâmica da Disciplina

- 10 aulas síncronas/
assíncronas;
- Ambiente Ava Moodle;
- Tutor On-line:
Denis S. Passos



Aulas Online -
ConfWeb RNP

Tutores Presenciais

Polo	Tutor(a)
Ananás	Veruska
Araguaína	Jefté
Arraias	Deusmar
Gurupi	Itamar Júnior
Miracema	Juscimar

Forma de Avaliação

- 10 aulas, sendo a última Exame Final;
 - Semanalmente, há uma atividade avaliativa, valendo 1 ponto, para os 7 encontros;
 - A participação nos fóruns da disciplina vale 1 ponto;
 - Seminário valendo 2 pontos.
- Se Nota Aluno (NA)
 - $\geq 7,0$ APROVADO DIRETO;
 - $< 4,0$ REPROVADO DIRETO;
 - $4,0 \leq NA \leq 6,9$
então EXAME FINAL

Forma de Avaliação

- Como funciona o Exame Final (EF) ?
- Uma prova valendo 10 pontos;
- Toma-se a Nota Final (NF):
$$NF = (NA + EF) / 2$$
- Se NF maior ou igual a 5 o aluno está **APROVADO COM EXAME**, caso contrário está **REPROVADO COM EXAME**.



Bibliografia e Metodologia

Básica

- 1 - PREISS, Bruno R. Estrutura de dados e algoritmos: padrões de projetos orientados a objetos com java. Rio de Janeiro: Elsevier, 2001.
- 2 - EDELWEISS, N; GALANTE, R. Estrutura de Dados. Volume 18 da Série Livros Didáticos Informática UFRGS. Bookman, 2009.
- 3 - CORMEN, T. H. Algoritmos: Teoria e Prática. Campus, 2002.

Bibliografia e Metodologia

Complementar

- 1 – SZWARCFTTER, J.L.; MAKENZON, L. Estruturas de Dados e seus Algoritmos. Rio de Janeiro: LTC, 2004.
- 2 - AGUILAR, Luis Joyanes. Programação em C++: algoritmos, estruturas de dados e objetos. 2a. ed. São Paulo: McGraw-Hill, 2008.
- 3 - PREISS, Bruno R. Estrutura de Dados e Algoritmos – Padrões de Projetos orientados a objetos com Java. Rio de Janeiro: Campus, 2000.

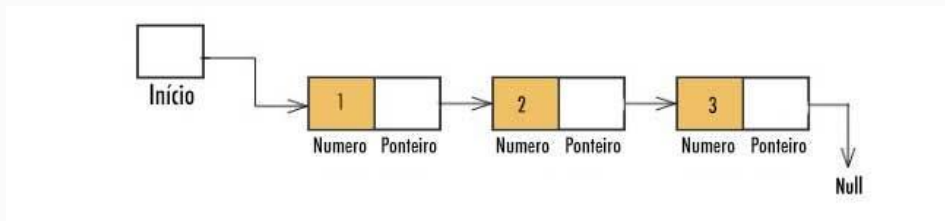


O que são Estruturas de Dados?

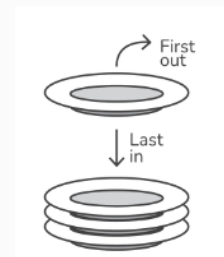
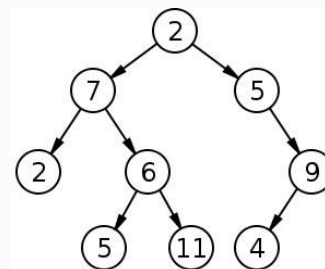
- Uma **estrutura de dados** é uma maneira de armazenar e relacionar conjuntos de informações de forma organizada em um computador de modo que possam ser usados eficientemente, facilitando sua busca e modificação;
- Existe um conjunto de estruturas de dados que vamos chamar de "**clássicas**".

Estruturas de Dados Clássicas

- Vetores;
- Listas;
- Pilhas;
- Filas;
- Árvores.



	Col. 0	Col. 1	Col. 2	Col. 3	Col. 4
Linha 0	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)
Linha 1	(1,0)	(1,1)	(1,2)	(1,3)	(1,4)
Linha 2	(2,0)	(2,1)	(2,2)	(2,3)	(2,4)
Linha 3	(3,0)	(3,1)	(3,2)	(3,3)	(3,4)



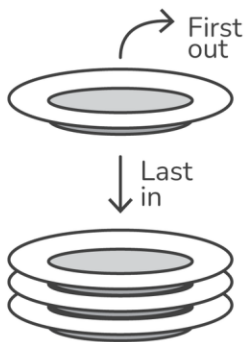


Estruturas de Dados no Mundo Real

- Onde podemos observar essas estruturas de dados no nosso dia-a-dia?

Estruturas de Dados no Mundo Real

- Onde podemos observar essas estruturas de dados no nosso dia-a-dia?



the LIFO principle
last in - first out

Pilha



Fila

Estruturas de Dados no Mundo Real

- Onde podemos observar essas estruturas de dados no nosso dia-a-dia?

LISTA DE COMPRAS
PARA AJUDAR NAS SUAS DECISÕES FINANCEIRAS INTELIGENTES!

ALIMENTOS	HIGIENE E LIMPEZA
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

FRUTAS E LEGUMES	UTILIDADES
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>

BEBIDAS
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>
<input type="checkbox"/>

Lista



Árvore



Estruturas de Dados

- Cada uma dessas estruturas de dados é útil para uma situação particular;
- Cada estrutura possui uma política (regras de controle, inserção/remoção) distinta das outras;
- Entender, manter e aplicar essas políticas são úteis para muitas aplicações;
- Por exemplo, o uso de pilhas tem grande aplicação na construção de compiladores;
- Por exemplo, ainda, algoritmos de árvores tem um dos melhores desempenhos entre os algoritmos de busca;

Vamos pôr a mão na massa?

- Em primeiro lugar vamos nos ambientar:
- Vamos implementar todas as estruturas que formos estudar em Python. Portanto, instale Python em sua máquina !



Jupyter Notebook



Google Colab



PyCharm

Ciclo de Desenvolvimento de um Programa

- Por enquanto acessem o Colab do Google enquanto escolhem sua IDE de desenvolvimento.
- <https://colab.google/>

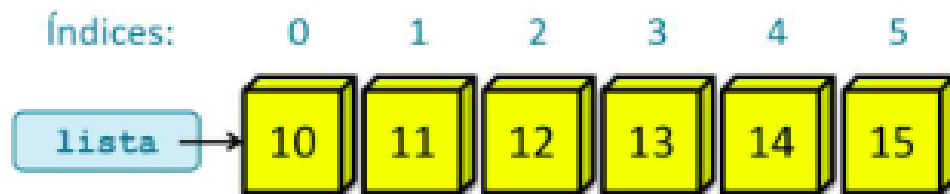


Introdução às Listas

Listas em Python

O que são Listas?

- Uma lista é uma estrutura de dados organizada de forma sequencial;
- A lista difere do vetor que é estático (e tem tamanho definido na sua criação), enquanto a lista pode ter seu tamanho modificado dinamicamente;



Características das Listas em Python

- A lista é identificada por um **único nome**;
- Cada elemento da lista é referenciado por um **índice**;
- Os elementos da lista **podem** ser **modificados**;
- O nome da lista é uma referência (ponteiro) para o primeiro elemento;

Notas →	6.1	2.3	9.4	5.1	8.9	9.8	10	7.0	6.3	4.4
Posição:	0	1	2	3	4	5	6	7	8	9



Características das Listas em Python

- A lista é uma estrutura de dados que possui uma política bem flexível, permitindo efetuar inserções no seu começo, no final ou até mesmo alterar de um elemento arbitrário no interior da lista;
- A seguir apresentaremos algumas funções úteis para manipulação de listas em Python.

Funções para Manipulação de Listas

- Para criar uma lista é necessário declarar uma variável e passar os elementos da lista entre parênteses, por exemplo.
 - ListNum = [75, 93, 25, 17, 42]

Esse comando resultará na seguinte lista:

ListNum =

75	93	25	17	42
-----------	-----------	-----------	-----------	-----------

Funções para Manipulação de Listas

- `list.append(x)` - Adiciona um item ao fim da lista.

- Exemplo de uso:

ListNum =

75	93	25	17	42
-----------	-----------	-----------	-----------	-----------

ListNum.append(55)

1. Como fica a lista **ListNum** depois da execução desta operação?
2. Adicione mais um elemento e mostre como ficou a lista **ListNum**.

Funções para Manipulação de Listas

- `list.insert(i, x)`
- Insere um item em uma dada posição. O primeiro argumento é o índice do elemento antes do qual será feita a inserção.
 - Exemplo de uso:

ListNum =

75	93	25	17	42
-----------	-----------	-----------	-----------	-----------

ListNum.insert(2,85)

1. Como fica a lista **ListNum** depois da execução desta operação?
2. Adicione mais elementos nas **posições 3 e 4** e apresente **ListNum**.

Funções para Manipulação de Listas

- **len(list)** - Retorna o tamanho da lista

- Exemplo de uso:

ListNum =	75	93	25	17	42
------------------	-----------	-----------	-----------	-----------	-----------

len(ListNum)

1. Adicione elementos em **ListNum** e veja como o tamanho varia.

Funções para Manipulação de Listas

- `list.remove(x)` - Remove o primeiro item encontrado na lista cujo valor é igual a `x`. Se não existir valor igual, uma exceção `ValueError` é levantada.

- Exemplo de uso:

ListNum =

75	93	25	17	42
-----------	-----------	-----------	-----------	-----------

ListNum.remove (75)

1. Adicione elementos 18, 93 e 24 ao final de **ListNum**, após isso, remova o elemento 93. Como ficou ListNum?

Funções para Manipulação de Listas

- `list.count(x)` - Devolve o número de vezes em que `x` aparece na lista.
 - Exemplo de uso:

ListNum =

75	93	75	17	75
-----------	-----------	-----------	-----------	-----------

ListNum.count(75)

Funções para Manipulação de Listas

- `list.index(x[, start[, end]])` - Devolve o índice base-zero do primeiro item cujo valor é igual a x, levantando `ValueError` se este valor não existe.

- Exemplo de uso:

ListNum =

28	93	22	17	22
-----------	-----------	-----------	-----------	-----------

ListNum.index(22)



Exercícios de Manipulação de Listas

- Escreva o código abaixo

```
fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']  
fruits.count('apple')
```

Exercícios de Manipulação de Listas

- Escreva o código abaixo

```
fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']  
#fruits.count('apple')  
fruits.count('tangerine')
```



Exercícios de Manipulação de Listas

- Escreva o código abaixo

```
fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']  
#fruits.count('apple')  
#fruits.count('tangerine')  
fruits.index('banana')
```

Exercícios de Manipulação de Listas

- Escreva o código abaixo

```
fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']  
#fruits.count('apple')  
#fruits.count('tangerine')  
#fruits.index('banana')  
fruits.index('banana', 4)
```


Exercícios de Manipulação de Listas

- Escreva o código abaixo

```
fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']  
#fruits.count('apple')  
#fruits.count('tangerine')  
#fruits.index('banana')  
#fruits.index('banana', 4)  
fruits.append('grape')
```

Exercícios de Manipulação de Listas

- Escreva o código abaixo

```
fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']  
#fruits.count('apple')  
#fruits.count('tangerine')  
#fruits.index('banana')  
#fruits.index('banana', 4)  
#fruits.append('grape')  
fruits.insert(2, 'melon')
```

Exercício 1

Considere as seguintes listas:

$L1 = [2, 3, 8, 10]$

$L2 = [1, 0, 56, 29]$

$L3 = L1 + L2$

Pergunta, qual a configuração da lista L3 ?

Exercício 2

Considere o código abaixo:

```
L1 = [2, 3, 8, 10]
```

```
L2 = L1
```

```
L2[1] = 5
```

```
L2[2] = 5
```

```
print( L1 )
```

Pergunta, qual é a saída apresentada pelo comando print ?



Tarefas Semanais

- Refazer Exercícios da Aula;
- Responder Questionário Avaliativo (vale 1.0 ponto);
- Responder Fórum;
- Fazer leitura recomendada;
- Estudar Listas, Filas e Pilhas em Python.



Conclusão e Próxima Aula

- **Aula de Hoje:**
 - **Apresentação da Disciplina;**
 - **Conceitos Básicos de Estruturas de Dados;**
 - **Prática de Programação.**
- **Próxima Aula:**
 - **Pilhas em Python;**
 - **Listas como Pilhas.**

Algoritmos e Estruturas de Dados

Aula 1

Prof Dr Tanilson Dias dos Santos

Universidade Aberta do Brasil – UAB
Universidade Federal do Tocantins - UFT