

INSTITUTO FEDERAL DE ALAGOAS – IFAL / CAMPUS MACEIÓ

Curso Bacharelado em Sistemas de Informação

Disciplina de Estruturas de Dados

Prof. MSc. Ricardo Nunes

Ricardo (arroba) ifal.edu.br

Listas

Objetivos

1. Entender o conceito de Listas
2. Aprender a criar Listas de dados complexos

Conteúdo

1. Problemática
2. Listas em C e Python
3. Tempo gasto com execução
4. TAD - Listas
5. Alcoação Dinâmica

Problema

- Dado o tipo Candidato, como desenvolver um tipo para armazenar coleções de candidatos?
- Como deve ser a interface desse tipo, ou seja, que operações são executadas sobre os dados?
- Adiciona um dado candidato no fim da Lista.
- Adiciona um dado candidato em uma dada posição.
- Pega o candidato de dada posição.
- Remove o candidato de dada posição.
- Verifica se um dado candidato está armazenado.
- Informa o número de alunos armazenados.

Como seria em C?

Lembre-se da manipulação de String em C!

□

Como seria em Python?

- Primeiro os testes...
- Depois o código!
- Como seria inserir no final da lista?
- Como seria inserir em uma posição qualquer da lista ?

ver

- lista_candidato1_test.py
- lista_candidato1.py

Tempo gasto

- Qto tempo é gasto para inserir um elemento?

* quanto mais candidatos forem inseridos pior será o desempenho desta função
* consumo é linear

- como melhorar?

* guardando o tamanho da lista

* mas como deixar junto lista e tamanho?
* criando um novo tipo de dados...

ver:

- lista_candidato2_test.py
- lista_candidato2.py
- classificacao_jogo.py
- classificacao_jogo_main.py
- lista_candidato3_test.py #INCOMPLETO
- lista_candidato3.py #INCOMPLETO

Como implementar em Python as demais funções?

- Como seria pegar da posição x?
- Como seria remover da posição x?
- Como seria verificar se a lista tem um dado elemento?
- Como seria informar o tamanho da lista?

Alocação Dinâmica

- Qual o problema da nossa abordagem até agora?

* Desperdício de alocação! * Funciona somente para lista de candidatos.

Como resolver isso?

* Usando estruturas de dados da linguagem, neste caso, `list` em Python * Implementando as próprias estruturas

Código fonte

<https://replit.com/@RicardoRubens/listas-21>

Para estudar

- Módulo 1 do livro [4] <http://educapes.capes.gov.br/handle/capes/176522>
- Seções 1.13 do Livro [4] https://panda.ime.usp.br/pythonds/static/pythonds_pt/04-Recursao/toctree.html

Referências

1. Tradução do livro *How to Think Like a Computer Scientist: Interactive Version*, de Brad Miller e David Ranum. link: <https://panda.ime.usp.br/pensepy/static/pensepy/index.html>
2. Allen Downey, Jeff Elkner and Chris Meyers. *Aprenda Computação com Python 3.0*. link: <https://chevitarese.files.wordpress.com/2009/09/aprendacomputaocompython3k.pdf>
3. SANTOS, A. C. *Algoritmo e Estrutura de Dados I*. 2014. Disponível em <http://educapes.capes.gov.br/handle/capes/176522>
4. SANTOS, A. C. *Algoritmo e Estrutura de Dados II*. 2014. Disponível em 2014. Disponível em <https://educapes.capes.gov.br/handle/capes/176557>
5. Tradução do livro [5] *Problem Solving with Algorithms and Data Structures using Python* de Brad Miller and David Ranum. link: https://panda.ime.usp.br/pythonds/static/pythonds_pt/index.html
6. Brad Miller and David Ranum. *Problem Solving with Algorithms and Data Structures using Python* link: https://panda.ime.usp.br/pythonds/static/pythonds_pt/index.html
7. Caelum. *Algoritmos e Estruturas Dados em Java*. Disponível em <https://www.caelum.com.br/download/caelum-algoritmos-estruturas-dados-java-cs14.pdf>

That's all Folks
