



www.python.pro.br

Roteiro - Vetor e Lista

Vetor

Lista

List, Tuple e String - Python

Operações não mutáveis

Operações Mutáveis

Exercício

Vetor

Espaço contíguo em memória

Índices, Endereços e RAM

Prós:

- Armazenamento eficiente

- Acesso a elementos em $O(1)$

Contras:

- Necessidade de saber o tamanho*

Vetor - Python array

Só serve para "tipos primitivos"

Não muito utilizado *

Vetor - Python array

Code	C Data Type	Typical Number of Bytes
'b'	signed char	1
'B'	unsigned char	1
'u'	Unicode char	2 or 4
'h'	signed short int	2
'H'	unsigned short int	2
'i'	signed int	2 or 4
'I'	unsigned int	2 or 4
'l'	signed long int	4
'L'	unsigned long int	4
'f'	float	4
'd'	float	8

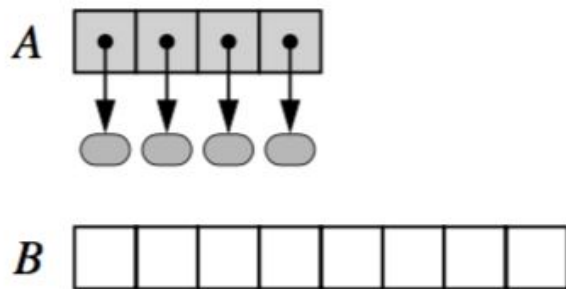
Table 5.1: Type codes supported by the array module.

Lista

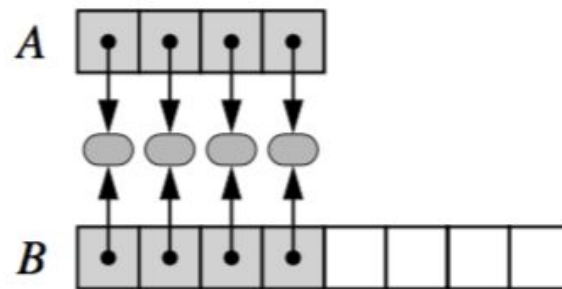
Também chamado array dinâmico

Permite inserção de elementos

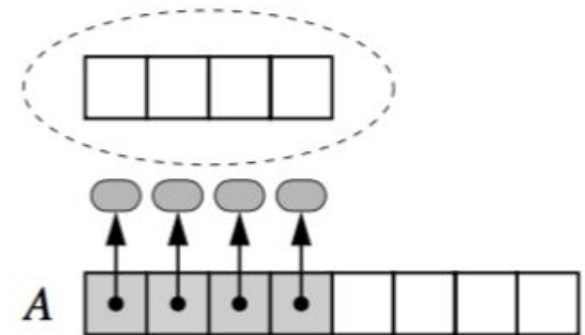
Implementado com vetor por baixo dos panos



(a)



(b)



(c)

List, Tuple e String - Python

Também chamados de sequências

Tupla é uma lista imutável

Mais eficiente em termos de memória

Quando usar uma outra?

Retorno de múltiplos elementos

Desempacotamento

String também é imutável

Tratada como tupla *

Operações não mutáveis

Operation	Running Time
<code>len(data)</code>	$O(1)$
<code>data[j]</code>	$O(1)$
<code>data.count(value)</code>	$O(n)$
<code>data.index(value)</code>	$O(k + 1)$
<code>value in data</code>	$O(k + 1)$
<code>data1 == data2</code> (similarly <code>!=</code> , <code><</code> , <code><=</code> , <code>></code> , <code>>=</code>)	$O(k + 1)$
<code>data[j:k]</code>	$O(k - j + 1)$
<code>data1 + data2</code>	$O(n_1 + n_2)$
<code>c * data</code>	$O(cn)$

Operações Mutáveis

Operation	Running Time
<code>data[j] = val</code>	$O(1)$
<code>data.append(value)</code>	$O(1)^*$
<code>data.insert(k, value)</code>	$O(n - k + 1)^*$
<code>data.pop()</code>	$O(1)^*$
<code>data.pop(k)</code> <code>del data[k]</code>	$O(n - k)^*$
<code>data.remove(value)</code>	$O(n)^*$
<code>data1.extend(data2)</code> <code>data1 += data2</code>	$O(n_2)^*$
<code>data.reverse()</code>	$O(n)$
<code>data.sort()</code>	$O(n \log n)$

*amortized

Exercício

Criar função que retorne min e max de uma sequência

Não pode usar biblioteca (max e min)

Deve ser recursivo (sem for ou while)

Deve informar via docstring qual é a complexidade de tempo e espaço

Enviar link do código no Github

Form enviado via fórum *

Roteiro - Vetor e Lista

Vetor

Lista

List, Tuple e String - Python

Operações não mutáveis

Operações Mutáveis

Exercício

Obrigado

renzo@python.pro.br
@renzoprobr

