

Lista de Computação I (Python)

DCC / UFRJ

Pedro Maciel Xavier (monitor)
pedromxavier@poli.ufrj.br

25 de dezembro de 2019

Introdução

Essa lista de exercícios ainda se encontra em desenvolvimento. A intenção é que ela tenha um gabarito bem aberto, deixando muito das respostas para a criatividade do aluno. As questões são, em geral, grandes para se resolver e podem necessitar de alguma pesquisa adicional. Elas tem estrelinhas ★ indicando a dificuldade estimada. Alguns exercícios foram inspirados em outros propostos em materiais cujas fontes estão devidamente referenciadas no final. É importante você tire suas dúvidas e dê um retorno do que achou dos exercícios através do e-mail no cabeçalho.

Boa diversão!

Sumário

1	Condições (if , elif , else)	3
2	Números (int , float , complex)	4
2.1	Gerando números "aleatórios" [*]	4
3	<i>Strings</i> e Texto (str)	5
3.1	DNA [*]	5
3.2	[*]	5
4	Listas e Tuplas (list , tuple)	6
4.1	Eratóstenes de Cirene [*]	6
5	Funções e Recursividade (def , return)	7
6	Laços (for , while)	8
6.1	Sequência de Collatz [*]	8
7	Conjuntos e Dicionários (set , dict)	9
7.1	Mensagem cifrada [*]	9
8	Arquivos (with , open)	10

1 Condições (**if**, **elif**, **else**)

2 Números (**int**, **float**, **complex**)

2.1 - Gerando números "aleatórios" [★]

Gerar um numero aleatório é um pouco complicado em geral. Muitas formas de se fazer isso hoje em dia se baseiam em processos da natureza como fluidos em regime turbulento ou fenômenos quânticos. No entanto, é possível chegar perto disso com muito menos. Para começar, escolhemos uma potência de 2, como $N = 2^{16} = 65536$. Depois disso, escolhemos um número primo no meio do caminho, digamos, $p = 25773$. Por fim, um número ímpar menor que o número primo p , por exemplo, $b = 13849$.

Partimos de um número n_0 qualquer no intervalo $[0, 25536)$ e calculamos o sucessor da seguinte forma:

$$n_j = p \times n_{j-1} + b \mod N$$

Para obter um número "aleatório" distribuído de maneira uniforme no intervalo $[0, 1)$, basta calcular

$$x_j = \frac{n_j}{N}$$

Desafio: Faça o seu próprio gerador de números aleatórios!

3 *Strings* e Texto (**str**)

3.1 - DNA [★]

Uma sequência de DNA (*ácido desoxirribonucleico*) é composta por 4 nucleotídeos: **A**denina, **C**itosina, **G**uanina **T**imina. No RNA mensageiro (mRNA), a **T**imina está ausente, e dá lugar para a **U**racila. Quando um ribossomo realiza a transcrição de DNA em mRNA ele segue uma regra muito simples:

A	→	U
T	→	A
C	→	G
G	→	C

Faça uma função que traduza uma fita de DNA (**str**) como faria um ribossomo.

3.2 - [★ ★]

4 Listas e Tuplas (**list**, **tuple**)

4.1 - Eratóstenes de Cirene [* *]

5 Funções e Recursividade (**def**, **return**)

6 Laços (**for**, **while**)

6.1 - Sequência de Collatz [* *]

A sequência de Collatz é descrita recursivamente por:

$$f(n) \triangleq \begin{cases} 3n + 1, & \text{se } n \text{ for ímpar} \\ n \div 2, & \text{se } n \text{ for par} \end{cases}$$

Por exemplo, começamos com $n = 26$. Após sucessivas aplicações temos:

$$26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$$

Isso nos dá uma sequência com 11 números. 40 é maior que 26, mas sua sequência só teria 9 números.

Ainda não se sabe se todos os números induzem uma sequência que termina em 1. No entanto, até agora não foi encontrado um número sequer em que isso não tenha acontecido!

Deasfio: Faça uma função que calcule o comprimento da sequência gerada a partir de um número natural n qualquer.

7 Conjuntos e Dicionários (**set**, **dict**)

7.1 - Mensagem cifrada [* *]

8 Arquivos (**with**, **open**)