

Computação 1, 2021.2

Lista 3

Data limite para entrega: 21/12 às 18:00

Submeta suas soluções colocando os arquivos correspondentes na sua pasta do Google Drive*

Atualizada em 16/12, adiando data da entrega e corrigindo o nome da função `próximo_Fibonacci`

Lembre-se de escolher bons nomes para suas funções e variáveis, e de documentar seu código com *docstrings* (documentação de função) e comentários onde for apropriado.

Parte 1 — Obrigatória

Questão 1 (Fórmula de Binet). Sendo

$$\phi = \frac{1 + \sqrt{5}}{2} \quad e \quad \psi = \frac{1 - \sqrt{5}}{2},$$

o Teorema de Binet diz que o n -ésimo número de Fibonacci F_n (começando a contar a partir de $n = 0$, como faremos sempre) satisfaz:

$$F_n = \frac{\phi^n - \psi^n}{\sqrt{5}}. \quad (\star)$$

a. Implemente a fórmula de Binet em Python — a sua função deve receber n como entrada e retornar o valor do lado direito da igualdade (\star) dada acima.

b. Mostre que a fórmula de Binet **não é** uma boa forma de calcular números de Fibonacci com o computador. Faça isso escrevendo uma função de Python que retorna um número n , o valor *correto* de F_n , e o valor retornado pela sua função do item (a) acima, **diferente** de F_n .

c. Explique em suas palavras por que necessariamente a sua implementação da fórmula de Binet calcula o valor errado para algumas entradas.

*Link recebido por email em 24/11/2021 — o nome é parecido com <seu nome> - Comp 1 2021.2 - Submissões e Feedback.

Questão 2. Implemente uma função que receba dois números (`ints` ou `floats`) e retorne quantos números distintos há entre eles (inclusive contando com eles, se for o caso) que são números de Fibonacci. Por exemplo, com entradas 0 e 5.5, a resposta deve ser 5.

Questão 3. As funções `próximo_Fibonacci` (da aula 03p) e (talvez) a sua função da Questão 2 são um pouco ineficientes, pois calculam o valor de F_n para todos os valores de n até encontrarem o valor certo.

a. Substitua a chamada da função `Fibonacci` nestas funções para não calcular “do zero” os números de Fibonacci em `próximo_Fibonacci`. (Dica: o tempo para calcular `próximo_Fibonacci(10**2000)` é grande com o algoritmo visto em aula, mas pode ficar quase **2000 vezes menor!**)

b. Faça o mesmo para a sua função da Questão 2.

Parte 2 — Desafio opcional

Questão 4. Usando argumentos padrão, modifique a função `próximo_Fibonacci` para que ela comece a sua busca a partir de um certo índice dado. Assim, a sua função da Questão 2 pode usar a nova (e rápida) versão de `próximo_Fibonacci` a partir do valor de mínimo.

Que (outros) argumentos você precisa passar para a nova `próximo_Fibonacci`?