

Teste 3

Funções Recursivas

De 11 de Maio para 19 de Maio de 2022

Instruções: Este teste possui 3 partes.

- A primeira, composta pelas questões de 1 a 3, é para ser entregue ao final da aula.
- A segunda, composta pelas questões de 4 a 7, deve ser entregue até a próxima aula prática.
- A terceira, o desafio, é opcional, e sua entrega vai também até a próxima prática.

Parte 1 – Em sala

Questão 1. Escreva uma função que retorna o n -ésimo número de “Tri-bonnaci”, definido por

$$T_0 = 0, \quad T_1 = 1, \quad T_2 = 0, \quad T_{n+3} = T_{n+2} + T_{n+1} + T_n.$$

Questão 2. Dê uma documentação para a sua função, e inclua `doctests`.

Questão 3. Para quais valores de n temos $T_n > F_n$?

Parte 2 – Obrigatória

Questão 4. Escreva uma função `próximo_Fibonacci(n)` que retorna o menor número de Fibonacci que é *estritamente maior* do que n .

Questão 5. Escreva uma função `contar_Fibonacci(a,b)` que retorna quantos números de Fibonacci há no intervalo $[a, b]$. Note que 1 aparece duas vezes na sequência, mas deve ser contado como um único número.

Questão 6. Escreva a documentação e os `doctests` para as duas funções acima.

Questão 7. Escreva uma função que faz um teste *aleatório* (usando o módulo `random`) para as funções acima. Ela deve retornar `True` se a função passou no teste, e `False` caso contrário.

Parte 3 — Desafio opcional

Questão 8. Use o recurso `%time` do iPython (ou equivalente) para descobrir quanto tempo seu código leva para calcular `Fibonacci(n)` para vários valores de n , e compare com `Tribonacci(n)`. (vá *com calma!*)

Que explicação você dá para o tempo que estas funções levam para calcular?