

Tempo restante 0:56:05



Questão 10

Parcialmente correto

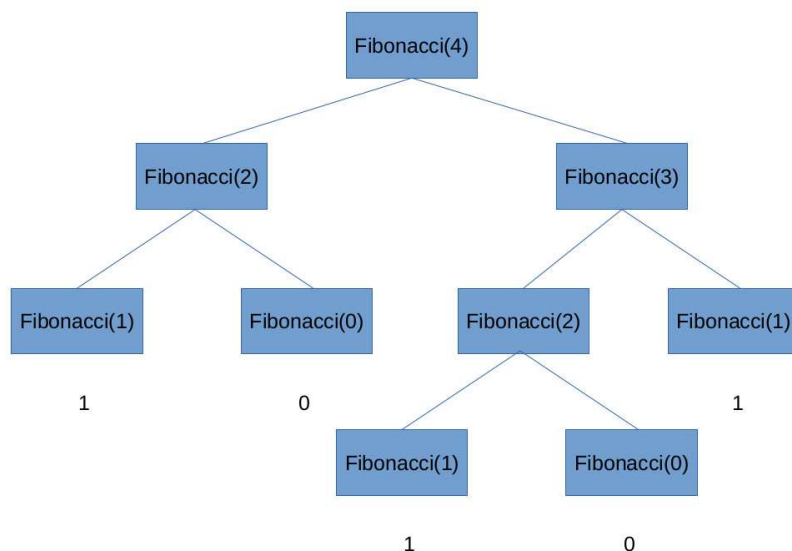
Atingiu 0,86 de 1,00

Fibonacci III

Na matemática, a Sequência de Fibonacci, é uma sequência de números inteiros, começando por 0, na qual, cada termo subsequente corresponde à soma dos dois anteriores. Os números de Fibonacci são, portanto, os números que compõem a seguinte sequência:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, ...

Após dominar o uso de [funções](#) recursivas envolvendo fibonacci, Charlinho ficou com uma pergunta na cabeça: "Quantas vezes cada chamada para a [função](#) fibonacci é realizada para cada número menor que n ?" Para ajudar a responder a essa pergunta, ele resolveu desenhar uma estrutura que parece uma árvore de ponta cabeça com as quantidades de chamadas, e ela ficou da seguinte forma:



Perceba que na árvore, Charlinho coloca no topo o valor que deseja calcular (Fibonacci de 4). Essa chamada gera outras duas chamadas para Fibonacci de 2 e 3. A chamada de Fibonacci de 2 gera as chamadas de Fibonacci de 1 e 0 enquanto a chamada de Fibonacci de 3 geram as chamadas de Fibonacci de 2 e 1. Por fim, a chamada de Fibonacci de 2 geram as chamadas de Fibonacci de 1 e 0.

Para não precisar sempre desenhar essa árvore, ele pediu a sua ajuda para implementar um [programa](#) que conte quantas vezes cada chamada da [função fibonacci](#) é realizada, para cada valor de entrada diferente da [função](#) recursiva. Observe que, ao elaborar seu [programa](#), você deve definir uma [função fibonacci](#) que recebe como [parâmetro](#) um número inteiro n fornecido da [entrada padrão](#), e considerá-la na resolução do problema.

Entrada

A entrada consiste em um único inteiro $0 \leq n \leq 30$ que indica a quantidade de termos da sequência de Fibonacci.

Saída

A saída deve conter uma linha com um inteiro indicando o n -ésimo termo da sequência de fibonacci, e em seguida outras $n + 1$ linhas com os quantitativos de vezes em que cada [função](#) é chamada para um determinado valor menor ou igual a n , conforme os exemplos.

Observações

- No terceiro exemplo de teste, o quarto termo da sequência de fibonacci é 3. Assim "Termo: 3". Para calcular o fibonacci de 4, calculam-se: 2 vezes o fibonacci de 0, 3 vezes o fibonacci de 1, 2 vezes o fibonacci de 2, 1 vez o fibonacci de 3 e 1 vez o fibonacci de 4.

For example:

Input	Result
0	Termo: 0 Quantidades: fibonacci(0) - 1
1	Termo: 1 Quantidades: fibonacci(0) - 0 fibonacci(1) - 1



Input	Result
4	Termino: 3 Quantidades: fibonacci(0) - 2 fibonacci(1) - 3 fibonacci(2) - 2 fibonacci(3) - 1 fibonacci(4) - 1

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 listfib =[0,0,0,0,0]
2 listfib += [0,0,0,0,0]
3 listfib += [0,0,0,0,0]
4 listfib += [0,0,0,0,0]
5 listfib += [0,0,0,0,0]
6 listfib += [0,0,0,0,0]
7 #print(len(listfib))
8 def fib(n):
9     if n == 0:
10         listfib[0] += 1
11         return 0
12     elif n == 1:
13         listfib[1] += 1
14         return 1
15     else:
16         listfib[n] += 1
17         return fib(n-1) + fib(n-2)
18 n = int(input())
19 print(f"Termino: {fib(n)}")
20 print("Quantidades:")
21 for i in range(len(listfib)):
22     if not(1 in listfib[i]):
23         break
24     else:
25         print(f"fibonacci({i}) - {listfib[i]}")
26
27

```



PRECHECK **VERIFICAR**

	Input	Expected	Got	
✓	0	Termino: 0 Quantidades: fibonacci(0) - 1	Termino: 0 Quantidades: fibonacci(0) - 1	✓
✓	1	Termino: 1 Quantidades: fibonacci(0) - 0 fibonacci(1) - 1	Termino: 1 Quantidades: fibonacci(0) - 0 fibonacci(1) - 1	✓
✓	4	Termino: 3 Quantidades: fibonacci(0) - 2 fibonacci(1) - 3 fibonacci(2) - 2 fibonacci(3) - 1 fibonacci(4) - 1	Termino: 3 Quantidades: fibonacci(0) - 2 fibonacci(1) - 3 fibonacci(2) - 2 fibonacci(3) - 1 fibonacci(4) - 1	✓



	Input	Expected	Got	
✓	12	Termo: 144 Quantidades: fibonacci(0) - 89 fibonacci(1) - 144 fibonacci(2) - 89 fibonacci(3) - 55 fibonacci(4) - 34 fibonacci(5) - 21 fibonacci(6) - 13 fibonacci(7) - 8 fibonacci(8) - 5 fibonacci(9) - 3 fibonacci(10) - 2 fibonacci(11) - 1 fibonacci(12) - 1	Termo: 144 Quantidades: fibonacci(0) - 89 fibonacci(1) - 144 fibonacci(2) - 89 fibonacci(3) - 55 fibonacci(4) - 34 fibonacci(5) - 21 fibonacci(6) - 13 fibonacci(7) - 8 fibonacci(8) - 5 fibonacci(9) - 3 fibonacci(10) - 2 fibonacci(11) - 1 fibonacci(12) - 1	✓
✓	7	Termo: 13 Quantidades: fibonacci(0) - 8 fibonacci(1) - 13 fibonacci(2) - 8 fibonacci(3) - 5 fibonacci(4) - 3 fibonacci(5) - 2 fibonacci(6) - 1 fibonacci(7) - 1	Termo: 13 Quantidades: fibonacci(0) - 8 fibonacci(1) - 13 fibonacci(2) - 8 fibonacci(3) - 5 fibonacci(4) - 3 fibonacci(5) - 2 fibonacci(6) - 1 fibonacci(7) - 1	✓
✓	16	Termo: 987 Quantidades: fibonacci(0) - 610 fibonacci(1) - 987 fibonacci(2) - 610 fibonacci(3) - 377 fibonacci(4) - 233 fibonacci(5) - 144 fibonacci(6) - 89 fibonacci(7) - 55 fibonacci(8) - 34 fibonacci(9) - 21 fibonacci(10) - 13 fibonacci(11) - 8 fibonacci(12) - 5 fibonacci(13) - 3 fibonacci(14) - 2 fibonacci(15) - 1 fibonacci(16) - 1	Termo: 987 Quantidades: fibonacci(0) - 610 fibonacci(1) - 987 fibonacci(2) - 610 fibonacci(3) - 377 fibonacci(4) - 233 fibonacci(5) - 144 fibonacci(6) - 89 fibonacci(7) - 55 fibonacci(8) - 34 fibonacci(9) - 21 fibonacci(10) - 13 fibonacci(11) - 8 fibonacci(12) - 5 fibonacci(13) - 3 fibonacci(14) - 2 fibonacci(15) - 1 fibonacci(16) - 1	✓

Your code failed one or more hidden tests.

Parcialmente correto

Notas para este envio: 0,86/1,00.

