EMB5016 – CÁLCULO NUMÉRICO
LISTA DE EXERCÍCIOS 3
LAÇOS E SÉRIES NUMÉRICAS

PROFESSOR: LUIZ GUSTAVO CORDEIRO

Instruções

Você deve criar todas as funções abaixo em um arquivo intitulado `lista_3.py`, e enviar este arquivo no VPL apropriado. Não esqueça de avaliar sua nota!

Você deve resolver os exercícios com processos básicos: soma/subtração, multiplicação/divisão, chamadas de entradas de listas, laços, etc. Não é permitido utilizar funções matemáticas ou métodos de controle de dados mais complexos padrão do Python (e.g. `**`, `pow`, `reverse`) nestes exercícios. Mas, se você quiser, pode fazer sua própria implementação deles.

Exercício 1. As funções $\sin(x)$ e $\cos(x)$ têm séries de Taylor

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} \qquad \text{e} \qquad \cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}.$$

Implemente essas séries em funções

`sin(x)` e `cos(x)`,

com argumento `x` float de precisão dupla. (O erro absoluto não pode passar de 2^{-30}.)

Exercício 2. A série de Taylor baseada em 1 da função $x^{-1/2}$ é

$$x^{-1/2} = \sum_{n=0}^{\infty} (-1)^n \frac{(2n)!}{(n!)^2 4^n} (x-1)^n,$$

e converge sempre que $0 < x \leq 2$.

Implemente essa série

`invsqrt(x)`

que toma um número positivo x *qualquer* e retorna $x^{-1/2}$.

Exercício 3. Crie duas funções:

`reverter(L)` e `reversa(L)`

que tomam como argumento uma lista `L`.

A função `reverter(L)` deve reverter e sobrescrever a lista `L`

A função `reversa(L)` deve reverter a lista `L` e retornar o resultado em uma nova lista, sem alterar `L`.

Por exemplo:

```
L=[1,2,3]        #Define a lista L
A=reversa(L)     #L=[1,2,3], A=[3,2,1]
reverte(L)       #L=[3,2,1], A=[3,2,1]
```

O objetivo deste exercício é que você se atente para o modo com que *nomes* e *valores* são tratados em Python. Em termos simples, você pode pensar que uma lista em Python é implementada do mesmo modo que em `C`: como um ponteiro para uma posição de memória. Para ler mais, clique aqui.

Exercício 4. Crie uma função

`ordena(L)`

que recebe como argumento uma lista de números `L` e retorna uma nova lista com as entradas de `L` ordenadas (em ordem crescente).[1]

Exercício 5. O seguinte pseudocódigo calcula $\log_2(x)$ de modo recursivo para qualquer $x > 0$:

- Primeiro, determine o número inteiro ℓ_0 tal que $2^{\ell_0} \leq x < 2^{\ell_0+1}$. Defina $x_0 = \dfrac{x}{2^{\ell_0}}$.
- Para $n \geq 0$, defina x_{n+1} e ℓ_{n+1} em casos:
  - Se $x_n^2 < 2$, defina $x_{n+1} = x_n^2$ e $\ell_{n+1} = \ell_n$.
  - Se $x_n^2 \geq 2$, defina $x_{n+1} = \dfrac{x_n^2}{2}$ e $\ell_{n+1} = \ell_n + 2^{-(n+1)}$.

Com esta definição, pode-se mostrar que

$$\log_2(x) = \ell_n + 2^{-n}\log_2(x_n) \qquad \text{e} \qquad \log_2(x) - 2^{-n} < \ell_n \leq \log_2(x) \qquad \text{para todo } n,$$

e portanto ℓ_n converge para $\log_2(x)$ quando $n \to \infty$.

Implemente o pseudocódigo acima e defina uma função

`logaritmo_base2(x)`

que recebe um número real (float) $x > 0$ e uma aproximação $\log_2(x)$. (A tolerância mínima aceitável será de 2^{-30}.)

Exercício 6. A *Conjectura de Collatz* – também conhecida como *Conjectura* $3x+1$ – é uma das conjecturas em abertas mais famosas da Matemática moderna. Dado um numero natural x_0, defina a sequência $\{x_n\}_n$ por

$$x_n = \begin{cases} 3x_n + 1 & \text{, se } x_n \text{ é ímpar} \\ x_n/2 & \text{, se } x_n \text{ é par,} \end{cases}$$

que chamamos de *sequência de Collatz baseada em* x_0. A Conjectura de Collatz afirma que, qualquer que seja x_0, tem-se que $x_n = 1$ para algum n.

Por exemplo:

- Se $x_0 = 4$, então $x_2 = 1$, e a conjectura se verifica.
- Se $x_0 = 3732423$, então $x_{598} = 1$, e a conjectura se verifica.

Crie uma função

`Collatz(x0)`

que recebe um número inteiro `x0` e retorna o menor índice j tal que a sequência de Collatz $\{x_n\}_n$ baseada em `x0` que satisfaz $x_j = 1$.

Por exemplo, `Collatz(4)` deve retornar `2`, e `Collatz(3732423)` deve retornar `596`.

Departamento de Engenharias da Mobilidade – Universidade Federal de Santa Catarina, Joinville, SC, Brazil

Email address: luiz.cordeiro@ufsc.br

[1]Procure e implemente algum algoritmo de ordenação: https://en.wikipedia.org/wiki/Sorting_algorithm. Pode parecer um problema simples, mas é extremamente importante criar estruturas de dados ordenadas na prática, para fazer sistemas de armazenamento e busca eficientes.