

Programação II

Trabalho de casa 2

Somas de raízes

Entrega a 10 de abril de 2017

Sabemos que este tipo de problema vos atormenta, que sonham todos os dias com somas de raízes quadradas. Qual a soma das raízes quadradas dos primeiros 100 números positivos? E quantas raízes preciso de somar antes de atingir o número 1000? Felizmente que a disciplina de Programação 2 está aqui para vos ajudar a dormir melhor.

1. Começamos pelo primeiro problema, a escrita em Python da função $\text{soma_raizes}(n) = \sum_{i=1}^n \sqrt{i}$. Resolvemos este problema com uma simples expressão, usando as funções de ordem superior da linguagem Python e sem precisarmos de recorrer a recursão ou iteração.
2. Para o segundo problema estamos interessado na lista das somas das primeiras n raízes quadradas, isto é, na função $\text{lista_somas_raizes}(n) = [\sqrt{1}, \sqrt{1} + \sqrt{2}, \sqrt{1} + \sqrt{2} + \sqrt{3}, \dots, \sum_{i=1}^n \sqrt{i}]$. Utilizando a função soma_raizes , conseguimos novamente resolver este problema utilizando as funções de ordem superior da linguagem Python e, claro, sem recorrer a recursão ou iteração.
3. Fazer a função $\text{lista_somas_raizes}$ chamar a função soma_raizes pode não ser a melhor ideia. É que passamos a vida a fazer os mesmos cálculos (na realidade quem passa a vida a refazer cálculos é o interpretador de Python). Imaginemos a lista $\text{lista_somas_raizes}(5)$. Quantas vezes calculamos $\sqrt{1}$? e $\sqrt{2}$? Conseguimos fazer melhor! Precisamos de uma espécie de **reduce** que devolve todos os estados intermédios do acumulador na forma de uma lista. A função `explora` recebe três argumentos: uma

função, uma lista e um elemento. Os três parâmetros da função têm exatamente o mesmo significado do que os da função `reduce`. Eis alguns exemplos:

```
>>> reduce(lambda acc, x: acc + x, [3, 5, 2, 1], 0)
11
>>> explora(lambda acc, x: acc + x, [3, 5, 2, 1], 0)
[0, 3, 8, 10, 11]
>>> reduce(lambda acc, x: x - acc, [3, 5, 2, 1], 0)
1
>>> explora(lambda acc, x: x - acc, [3, 5, 2, 1], 0)
[0, 3, 2, 0, 1]
>>> lista = [3, 4, 5, 3, 7, 9, 2, 1]
>>> reduce(lambda acc, x: x if x > acc else acc, lista, 2)
9
>>> explora(lambda acc, x: x if x > acc else acc, lista, 2)
[2, 3, 4, 5, 5, 7, 9, 9, 9]
```

Para a escrita da função `explora` podem usar iteração ou recursão.

4. E pronto. Na posse da função `explora` conseguimos escrever facilmente a função que obtém a lista da soma das raízes (alínea 2), sem nunca recorrer à função `soma_raizes`. Vamos apelidar esta nova função de `lista_somas_raizes_explora`.

Como curiosidade, se alguma vez vos perguntarem “de entre os primeiros 500 números positivos, quantos há cuja soma das raízes quadradas é inferior a 1000?”, já podem responder: “Exatamente estes:”

```
len(filter(lambda x: x < 1000, lista_somas_raizes(500)))
```

Para cada uma das 4 funções, inclua dentro do docstring:

1. Uma descrição da função incluindo o seu contrato (na forma de `Requires:` e `Ensures:`), tal como sugerido nas aulas.
2. A complexidade algorítmica da função e a sua justificação. Por exemplo:
 $O(n^2)$, onde n representa ..., porque ...

No caso da função `explora`, assuma que a função parâmetro é $\mathcal{O}(1)$.

Para além disso, o módulo em si deve estar equipado com uma descrição em formato `docstring`, tal como sugerido nas aulas. Não se esqueça de incluir o seu nome e número de estudante:

```
__author__ = Maria Lopes, 45638.
```

Tome em especial atenção os seguintes pontos.

- O nome do ficheiro que contém o vosso trabalho deve ser chamado `trabalho2_XXXXX.py` onde XXXXX é o vosso número de estudante.
- A primeira linha deste ficheiro deverá ser `# -*-coding:utf-8-*-`.
- O vosso código será testado por um processo automatizado. É indispensável que as vossas funções se chamem exatamente `soma_raizes`, `lista_somas_raizes`, `explora` e `lista_somas_raizes_explora` e que esperem exatamente o número de parâmetros enunciados acima.
- Este é um trabalho de resolução individual. Os trabalhos devem ser entregues no Moodle até às 23:59 do dia 10 de abril de 2017.
- Para este trabalho não precisam de entregar testes. Deve no entanto testar cada uma das 4 funções usando o método de partição do espaço de entrada. Nós iremos certamente testar as vossas funções utilizando este método.
- Os trabalhos de todos os alunos serão comparados por uma aplicação computacional. Releia com atenção a sinopse e lembre-se: “Alunos detetados em situação de fraude ou plágio, plagiadores e plagiados, ficam reprovados à disciplina (sem prejuízo de ser acionado processo disciplinar concomitante)”.