

Fundamentos de Algoritmos

Outras formas de repetição

Marco A L Barbosa

malbarbo.pro.br

Lembre-se de seguir o processo de projeto de funções e de usar o `mypy` e o `doctest` na etapa de verificação.

Começando

- 1) Qual é a forma geral da instrução “para cada no intervalo” e como ela é executada?
- 2) Quando devemos utilizar o “para cada no intervalo”?
- 3) Qual é a forma geral da instrução “enquanto” e como ela é executada?
- 4) Quando devemos utilizar o “enquanto”?
- 5) Quais são as quatro perguntas que precisamos responder para implementar uma função utilizando a abordagem incremental com o “enquanto”?

Praticando

- 6) Projete uma função que encontre as posições de todas as ocorrências de um nome em uma lista de nomes.
 - a) Esboce uma solução em duas etapas e depois implemente a função.
 - b) Faça uma implementação alternativa que use apenas uma repetição.
 - c) Faça uma implementação que use operações de sublista (suponha que as instruções de repetição não estão disponíveis).
 - d) Avalia e classifique as implementações em ordem de simplicidade.
- 7) Projete uma função que receba como entrada uma lista de números e uma posição e devolva uma nova lista sem o elemento da posição especificada. Para os itens a e b suponha que as operações de sublista não estão disponíveis.
 - a) Esboce uma solução em três etapas e depois implemente a função.
 - b) Faça uma implementação alternativa que use apenas uma repetição.
 - c) Faça uma implementação que use operações de sublista (suponha que as instruções de repetição não estão disponíveis).
 - d) Avalia e classifique as implementações em ordem de simples.
- 9) Rotacionar um arranjo n posições a esquerda significa mover os primeiros n elementos do arranjo para as últimas n posições do arranjo. Por exemplo, rotacionar o arranjo [5, 3, 4, 1, 7] duas posições a esquerda produz o arranjo [4, 1, 7, 5, 3]. Projete uma função que rotacione um arranjo n posições a esquerda.

- 10) Revise os exercícios da lista “Repetição e arranjos” e verifique quais podem ser implementadas utilizando o “enquanto” para deixar o código mais eficiente ou simples. Faça a implementação usando o “enquanto”.
- 11) Projete uma função que verifique se uma lista de números é dobrada, isto é, pode ser obtida pela concatenação de duas listas iguais (suponha que as operações de sublista não estão disponíveis).
- 12) Projete uma função que determine qual é a menor quantidade de elementos de uma lista que precisam ser somados (a partir do início da lista) para que a soma seja maior que um dado valor. Se não for possível atingir a soma desejada, a função deve devolver -1.
- 13) Projete uma função que receba como entrada uma string e um número natural n e crie uma nova string repetindo a string de entrada n vezes (suponha que o operador $*$ não está disponível).
- 14) Projete uma função que converta um número natural para uma string. Por exemplo, para o número 4561 a saída deve ser a string ‘4561’ (suponha que a função `str` não está disponível). Dica: faça divisões sucessivas por 10.
- 15) Projete uma função que converta uma string para um número natural. Por exemplo, para a string ‘4561’ a saída deve ser o número 4561 (suponha que a função `int` não está disponível).
- 16) Um número inteiro positivo n é perfeito se a soma dos seus divisores, exceto ele mesmo, é igual a n . Por exemplo, 6 é perfeito pois os divisores de 6 (exceto ele mesmo) são 1, 2 e 3 e a soma $1 + 2 + 3$ é igual a 6. O número 28 também é perfeito, verifique você mesmo! Projete uma função que verifique se um número inteiro positivo é perfeito.
- 17) Defina uma função que receba como parâmetro um valor n e calcule o valor aproximado de π usando os primeiros n termos da série

$$\pi = 4 \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} = 4 \left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots \right)$$

- 18) Projete uma função que receba um número inteiro positivo n , e crie a matriz identidade I_n , com n linhas e n colunas, com todos os elementos da diagonal principal (elementos com o mesmo índice) iguais a 1 e os demais elementos iguais a 0.
- 19) Projete uma função que encontre os índices de todas as linhas de uma matriz cuja a soma dos elementos é zero.
- 20) Projete uma função que encontre os índices de todas as colunas de uma matriz cuja a soma dos elementos é zero.
- 21) Projete uma função que verifique se uma matriz A é simétrica, isto é, para cada elemento a_{ij} da matriz, $a_{ij} = a_{ji}$.

Resolvendo problemas

- 22) A empresa Doce Natureza tem um aplicativo chamado frutamax, que vende frutas online. Você está participando do processo de recrutamento e tem um desafio inicial para fazer. Após um cliente fazer uma compra é preciso enviar um email com uma lista dos itens que o usuário comprou. A lista de itens deve ser escrita conforme o português, separando os itens com vírgula e usado “e” no final, como por exemplo: banana, maçã, morango e abacate. O desafio é projetar uma função que junte os itens de uma lista de strings nessa forma de texto em português.
- 23) O pessoal da Doce Natureza gostou da função que você escreveu, agora eles têm outro desafio para você. Ao final de cada mês eles precisam determinar as frutas que mais geraram receitas, eles têm uma lista com cada fruta e sua respectiva receita e o seu trabalho é fazer uma função que organize essa lista de maneira que as que mais geraram receitas apareçam primeiro. Você pode usar a seguinte estratégia, chamada de ordenação por seleção: encontrar a fruta com maior receita, removê-la da lista de entrada e adicioná-la no final da lista de saída e repetir o processo até que não tenha mais itens na lista de entrada.

- 24) A empresa que você trabalha sofreu uma falta de energia e agora é preciso recuperar os dados do backup. O primeiro passo é determinar o código dos clientes afetados. Em um primeiro momento foi obtido um arquivo (string) com o código de todos os clientes separados por vírgula. O seu trabalho agora é projetar uma função que gere uma lista dos códigos a partir dessa string. Por exemplo, para a string "512,12,145" a sua função deve gerar como resposta a lista [512, 12, 145].
- 25) Em um determinado jogo de construção de itens, cada item tem uma classe que varia de 1 a 10. Os item de classe 1 surgem conforme o jogador explorar os baús. Um item de classe 2 ou superior precisa ser construídos unindo dois itens da classe anterior. Por exemplo, para construir um item de classe 2 é necessário unir dois item de classe 1. Para construir um item de classe 10 é necessário unir dois item de classe 9. Projete uma função que receba como entrada um número n (de 1 a 10), e determine quantos itens de classe 1 são necessário para construir um item de classe n . Suponha que as únicas operações aritméticas disponíveis sejam a soma e a multiplicação.