Computação 1, 2021.1
Lista 5

Data limite para entrega: 24/8 às 23:59

Submeta suas soluções colocando os arquivos correspondentes na sua pasta do Google Drive*

Parte 1 — Obrigatória

Questão 1. Faça uma função que receba como entrada 3 inteiros `termo_inicial`, `razão` e `num_termos` e retorne um `range` correspondente à P.A. que tem termo inicial `termo_inicial`, razão `razão` e exatamente `num_termos` termos.

Questão 2. Em um alfabeto com apenas dois símbolos, `+` e `-`, digamos que uma expressão é *bem formada* se:

1. as quantidades de `+`s e `-`s na expressão são iguais;
2. à esquerda de cada símbolo `-`, a quantidade de `+`s é estritamente maior que a quantidade de `-`s.

Em outras palavras, uma expressão é bem formada sse começando com 0 e interpretando cada `+` como $+1$ e cada `-` como -1, chegamos ao final da expressão novamente com 0 e sem termos passado por números negativos em nenhum momento.

Faça uma função que receba uma string correspondendo a uma expressão neste alfabeto e retorne um booleano indicando se a expressão é bem formada ou não.

Questão 3. Em um alfabeto com apenas dois símbolos, `)` e `(`, digamos que uma expressão é *bem formada* se ela pode ser obtida de alguma expressão aritmética bem formada (de acordo com as regras aprendidas no ensino fundamental) apagando-se todos os símbolos que não são `)` nem `(`. Por exemplo, a expressão `(()(()))` é bem formada, pois poderia ter vindo da expressão aritmética `((2+3)/((1+1)*2))`, mas a expressão `(()` não é bem formada.

a. Dê a melhor caracterização que você puder para as expressões bem formadas nesta linguagem, ou seja, encontre a propriedade mais "interessante" que você conseguir de forma que uma expressão com os símbolos `)` e `(` é bem formada sse ela tem a propriedade dada.

*Link recebido por email em 19/7/2021 — o nome é parecido com `<seu nome> - Computação 1 2021.1 - Submissões e Feedback.`

b. Faça uma função que receba uma string correspondendo a uma expressão neste alfabeto e retorne um booleano indicando se a expressão é bem formada ou não.

Questão 4. Como vimos, na Cifra de César, cada letra na posição k é substituída pela letra na posição $k + e$, sendo e a chamada *chave de encriptação*.

a. Faça uma função para implementar a codificação da versão *multiplicativa* da Cifra de César, onde cada letra na posição k é substituída pela letra na posição $k \cdot e$ (voltando ao início do alfabeto sempre que necessário).

b. Na versão multiplicativa da cifra de César, nem toda chave de encriptação *funciona*. Na verdade, num alfabeto com n símbolos, uma chave e de encriptação funciona na cifra multiplicativa sse $\text{mdc}(e, n) = 1$ (você não precisa provar isso).

Dê um exemplo e explique o que acontece de errado no caso em que $\text{mdc}(e, n) \neq 1$.

c. Quando ela existe, uma chave de descriptação d correspondente a uma chave de encriptação e dada, na cifra multiplicativa com alfabeto de tamanho n, é qualquer inteiro d com a propriedade que $e \cdot d$ deixa resto 1 na divisão por n (você não precisa provar isso).

Faça uma função para deduzir uma chave de descriptação na versão multiplicativa da cifra de César *sempre que possível*: a função deve receber a chave de encriptação e e o tamanho n do alfabeto, e retornar uma chave de descriptação d correspondente, se existir, ou `False` em caso contrário. Você pode usar a sua implementação do Algoritmo de Euclides, da última lista de exercícios, ou a função `gcd` do módulo `math`, para calcular mdc.

d. Faça uma função para descriptar mensagens usando a cifra multiplicativa, sempre que possível. Sua função deve receber uma mensagem encriptada, a chave de encriptação que foi usada, e o alfabeto contendo todos os símbolos possíveis de serem usados, e deve retornar a mensagem descriptada, se possível, ou `False`, caso contrário.

Parte 2 — Desafio opcional

Questão 5. Sobre a Questão 3. Faça uma função que receba uma expressão bem formada `expr` e um natural `n` e retorne a menor subexpressão bem formada de `expr` que contenha (o símbolo na) a posição `n` da `expr` (começando a contagem de `0`, como sempre).

Por exemplo, com entradas `()((()(()()))` e `1`, a saída deve ser `()`, e com entradas `()((()(()()))` e `5`, a saída deve ser `((()())`.

Questão 6. Na Questão 4(c), é provavel que o seu algoritmo seja menos eficiente do que o possível. Implemente a seguinte ideia para encontrar o valor de d buscado (quando existe): modifique o algoritmo de Euclides (última lista de exercícios) para, recebendo como entrada naturais a e b, ao calcular um novo resto R a cada passo, também calcular valores inteiros x e y com a propriedade que $ax + by = R$. A ideia é notar que os novos valores de x e y buscados a cada passo podem ser obtidos a partir dos valores antigos usando uma fórmula simples, de forma que o processo todo pode ser efetuado eficientemente.