INSTITUTO FEDERAL
Catarinense
Campus Blumenau

Algoritmos

Bacharelado em Ciência da Computação
Prof. Paulo César Rodacki Gomes

Lista de exercícios - AER-Alg-26

Exercícios com conjuntos (sets) e dicionários em Python

O problemas desta lista devem ser resolvidos com o uso de conjuntos e dicionários em Python. Caso você utilize funções, para cada um dos exercícios, crie um programa main para executar suas funções. **OBS.: utilize as mesmas regras das listas anteriores para dar nomes aos arquivos.**

Questões de conjuntos:

1. **Caracteres únicos**. Escreva uma função Python para verificar se uma string possui caracteres únicos. Por exemplo, a string “azul" não repete letras, mas a string “ferramenta"possui letras repetidas. A função deve receber uma string e retornar True no primeiro caso (letras únicas) ou False caso contrário (letras repetidas). Você deve usar conjuntos para implementar a função.

2. **Diferença simétrica.** Escreva uma função Python que recebe dois conjuntos de números inteiros M e N e retorna uma lista com sua diferença simétrica em ordem ascendente. O termo diferença simétrica denota os valores que existem nos conjuntos M ou N, mas não existem em ambos os conjuntos simultaneamente. Por exemplo, para os conjuntos M = {2, 4, 5, 9} e N = {2, 4, 11, 12}, a resposta deveria ser [5, 9, 11, 12].

Questões de dicionários:

3. **Busca reversa**. Escreva uma função chamada buscaReversa, que encontra todas as chaves de um dicionário que estão mapeadas para um determinado valor. A função deve receber como parâmetros um dicionário e um valor para ser buscado no dicionário. A função deve retornar uma lista (possivelmente vazia) com as chaves encontradas. Escreva uma função main para demonstrar sua função. Note que a função deve funcionar independentemente de ela retornar múltiplas chaves, uma única chave, ou nenhuma chave.

4. **Código morse**. O código morse é um esquema de codificação de letras e números utilizando pontos e traços. Neste exercício você deve escrever um programa que usa um dicionário para armazenar o mapeamento de letras e números para código Morse. Voce deve representar os pontos com símbolo de ponto “.” e traços com sinal de subtração “-“. A tabela abaixo mostra o mapeamento de letras e números para código Morse. Seu programa deve ler uma mensagem do usuário e então deve traduzir cada letra e número para código Morse, deixando um espaço em branco entra cada caractere traduzido. O programa deve ignorar qualquer caracter que não seja letra ou número. Por exemplo, a mensagem `Hello, World!` Deve ser exibida da seguinte forma: `.... . .-.. .-.. --- .-- --- .-. .-.. -..`

Letter	Code	Letter	Code	Letter	Code	Number	Code
A	.-	J	.---	S	...	1	.----
B	-...	K	-.-	T	-	2	..---
C	-.-.	L	.-..	U	..-	3	...--
D	-..	M	--	V	...-	4	....-
E	.	N	-.	W	.--	5	.....
F	..-.	O	---	X	-..-	6	-....
G	--.	P	.--.	Y	-.--	7	--...
H	....	Q	--.-	Z	--..	8	---..
I	..	R	.-.	0	-----	9	----.

5. **Anagramas**. Duas palavras são anagramas se contiverem as mesmas letras, mas em ordens diferentes. Por exemplo: “amor" e “roma" são anagramas porque cada uma delas contém um "a", um "o", um "m" e um “r”. Crie uma função Python que recebe duas strings e retorna True se elas forem anagramas, ou False caso contrário.

6. **Anagramas novamente**. A noção de anagramas pode ser estendida para múltiplas palavras. Por exemplo: "William Shakespeare” e “I am a weakish speller” são anagramas se ignorarmos se as letras são maiúsculas e também os espaços. Crie uma nova versão da sua função do exercício anterior para verificar se duas frases são anagramas. Sua função deve desconsiderar se as letras são maiúsculas ou minúsculas, ignorar espaços e sinais de pontuação.

7. **Cartela de bingo**. Uma cartela de bingo é formada por 5 linhas e 5 colunas. As colunas são rotuladas com as letras B, I, N, G e O. Existem 15 números diferentes que podem aparecer abaixo de cada letra. Abaixo do B podem aparecer os números de 1 a 15; abaixo do I os números de 16 a 30, abaixo do N os números de 31 a 45 e assim por diante. Escreva uma função que cria uma cartela de bingo e a armazena em um dicionário. As chaves do dicionário são as letras B, I, N, G e O. Os valores devem ser listas de 5 números cada, que aparecem abaixo de cada letra na cartela. A função deve retornar o dicionário. Escreva uma segunda função que recebe o dicionário e exibe a cartela de bingo com as colunas rotuladas apropriadamente. Escreva um programa main que gere e exiba uma cartela de bingo usando suas funções.

8. **Bingo: verificando uma cartela vencedora**. Uma cartela de bingo vencedora deve conter uma linha (ou coluna ou diagonal) com 5 números que foram sorteados. Normalmente, nas cartelas de papel, os jogadores fazem um X sobre o número sorteado. Na sua implementação (no seu dicionário representando a cartela), vamos substituir por 0 o número que foi sorteado. Escreva uma função que receba como único parâmetro um dicionário representando uma cartela. Se a cartela contiver uma linha, coluna ou diagonal preenchida com zeros, a função deve retornar True. Caso contrário deve retornar False. Crie um programa que demonstre o funcionamento da sua solução criando e exibindo várias cartelas de bingo e indicando se cada uma é ou não é vencedora. Você deve mostrar pelo menos uma cartela com linha horizontal, uma com linha v ertical, uma com diagonal e por fim uma com alguns zeros cruzados, mas que não é vencedora. Você pode usar sua solução do problema anterior como ponto de partida para este exercício. **Dica**: como a cartela não tem números negativos, encontrar uma sequencia de 5 zeros é análogo a descobrir se a soma de uma sequencia de 5 números é igual a zero. Talvez você ache mais fácil fazer desse jeito.

9. **Jogo de Bingo**. Neste exercício você vai simular o jogo de Bingo para apenas uma cartela. Começa gerando uma lista de todas as chamadas válidas de bingo (B1 até O75). Depois que a lista estiver pronta, você pode embaralhar seus elementos chamando a função shuffle do módulo random do Python. Então seu programa deve ir utilizando os valores da lista para anunciar os números sorteados e zerar os números correspondentes na cartela até que ela contenha uma linha, coluna ou diagonal zerada. No seu programa principal, faça uma simulação de 1.000 partidas (sempre com uma nova cartela) e mostre o número mínimo, o máximo e a média de chamadas até que se tenha uma cartela vencedora. Utilize seu código dos dois exercícios anteriores e não se esqueça de criar novas funções sempre que voce identificar algum procedimento que pode ser melhor organizado dentro de uma função.