

Módulo Conjuntos e dicionários

Curso Estruturas de Dados e Algoritmos Expert

Prof. Dr. Nelio Alves

<https://devsuperior.com.br>

Lista de exercícios

Soluções:

<https://github.com/devsuperior/curso-eda/tree/main/conjuntos-dicionarios>

Problema "visitantes"

Fazer uma função que receba uma lista de registros de log de acesso a um website, e retorne a quantidade de usuários únicos que acessaram o site. Cada registro de log está no formato CSV (valores separados por vírgula) que contém: nome de usuário, momento de acesso e URL acessada.

Exemplo 1:

Entrada	Saída
["ana,2024-07-04T21:42:40.353283800Z,https://blog.com/login", "bob,2024-07-04T21:42:44.571283800Z,https://blog.com/news", "maria,2024-07-04T21:42:46.394283800Z,https://blog.com/shop", "ana,2024-07-04T21:42:50.026283800Z,https://blog.com/news"]	3

Exemplo 2:

Entrada	Saída
Arquivo visitantes-input.json no Github do curso (contém 100000 registros)	30680

Assinaturas:

Javascript:

```
function total(visitors)
```

Java:

```
public static int total(List<string> visitors)
```

C#:

```
public static int Total(List<string> visitors)
```

Python:

```
def total(visitors)
```

Problema "alunos"

Em um portal de cursos online, cada aluno possui um código único, representado por um número inteiro.

Cada instrutor do portal pode ter vários cursos, sendo que um mesmo aluno pode se matricular em quantos cursos quiser. Assim, o número total de alunos de um instrutor não é simplesmente a soma dos alunos de todos os cursos que ele possui, pois pode haver alunos repetidos em mais de um curso.

Você deve criar uma função que, dada a lista de alunos de todos os cursos de um instrutor, a função deve retornar a quantidade total de alunos deste instrutor.

Exemplo:

Entrada	Saída
<pre>[[15, 21, 80, 42], [21, 80, 47], [12, 21, 47, 35]]</pre>	7

Assinaturas:

Javascript:

```
function studentsCount(courses)
```

Java:

```
public static int studentsCount(List<List<Integer>> courses)
```

C#:

```
public static int StudentsCount(List<List<int>> courses)
```

Python:

```
def students_count(courses)
```

Problema "intersection" (ref: Leetcode intersection-of-two-arrays)

Empresas: Amazon, Google, Microsoft, Apple, Yandex, Criteo, Nvidia, etc.

Dados dois arrays num1 e num2, retorne um array contendo sua interseção. Cada elemento do array resultante deve ser único, e você pode apresentar os elementos em qualquer ordem.

Exemplo 1:

Entrada	Saída
<pre>{ "nums1": [1,2,2,1], "nums2": [2,2] }</pre>	[2]

Exemplo 2:

Entrada	Saída
<pre>{ "nums1": [4,9,5], "nums2": [9,4,9,8,4] }</pre>	<pre>[9,4] (ou [4,9])</pre>

Assinaturas:

Javascript:

```
function intersection(nums1, nums2)
```

Java:

```
public static int[] intersection(int[] nums1, int[] nums2)
```

C#:

```
public static int[] Intersection(int[] nums1, int[] nums2)
```

Python:

```
def intersection(nums1, nums2)
```

Problema "votacao"

Na contagem de votos de uma eleição, são gerados vários registros de votação contendo o nome do candidato e a quantidade de votos (formato .csv) que ele obteve em uma urna de votação. Você deve fazer uma função para receber todos registros de votação das urnas, e daí retornar um consolidado com os totais de cada candidato. O resultado pode ser mostrado em qualquer ordem.

Exemplo:

Entrada	Saída
["Alex Blue,15", "Maria Green,22", "Bob Brown,21", "Alex Blue,30", "Bob Brown,15", "Maria Green,27", "Maria Green,22", "Bob Brown,25", "Alex Blue,31"]	["Alex Blue,76", "Maria Green,71", "Bob Brown,61"]

Assinaturas:

Javascript:

```
function counting(votes)
```

Java:

```
public static List<String> counting(List<String> votes)
```

C#:

```
public static List<string> Counting(List<string> votes)
```

Python:

```
def counting(votes)
```

Problema "word-count"

Você deve fazer uma função que recebe um texto, e retorna o número de ocorrências de cada palavra no texto. Cada ocorrência deve ser representada por um objeto do tipo Rank, com os campos word e count. A saída da função deve ser uma lista de objetos tipo Rank. Você deve mostrar o resultado em letras minúsculas, ordenado da palavra mais frequente para a menos frequente, e as palavras com mesma frequência devem aparecer ordenadas alfabeticamente.

Antes de analisar as frequências das palavras, você deve aplicar a seguinte função **normalize** ao texto, para remover símbolos de pontuação, remover espaços adicionais e converter tudo para minúsculo:

Javascript:

```
function normalize(text) {  
    const words = text.replace(/^[^p{L}\p{N}\s]/gu, " ");  
    return words.replace(/\s+/g, " ").trim().toLowerCase();  
}
```

Java:

```
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public static String normalize(String text) {  
    Pattern pattern = Pattern.compile("[^\\p{L}\\p{N}\\s]", Pattern.UNICODE_CHARACTER_CLASS);  
    Matcher matcher = pattern.matcher(text);  
    String words = matcher.replaceAll(" ");  
    return words.replaceAll("\\s+", " ").trim().toLowerCase();  
}
```

C#:

```
using System.Text.RegularExpressions;  
  
static string Normalize(string text)  
{  
    string words = Regex.Replace(text, "[^\\p{L}\\p{N}\\s]", "", RegexOptions.Compiled);  
    return Regex.Replace(words, "\\s+", " ").Trim().ToLower();  
}
```

Python:

```
import re  
  
def normalize(text):  
    words = re.sub(r'^\w\s]', ' ', text, flags=re.UNICODE)  
    return re.sub(r'\s+', ' ', words).strip().lower()
```

Exemplo:

Entrada	Saída
O vento sussurra sons entre as árvores, sons que fazem animais correrem. A floresta e a natureza vibram com segredos e sons.	sons: 3 a: 2 e: 2 animais: 1 árvores: 1 as: 1 com: 1 correrem: 1 entre: 1 fazem: 1 floresta: 1 natureza: 1 o: 1 que: 1

	segredos: 1 sussurra: 1 vento: 1 vibram: 1
--	---

Assinaturas:

Javascript:

```
function wordCount(text)
```

Java:

```
public static List<Rank> wordCount(String text)
```

C#:

```
public static List<Rank> WordCount(string text)
```

Python:

```
def word_count(text)
```

Problema "two-sum" (ref: Leetcode two-sum)

Empresas: Amazon, Google. Microsoft, Meta, Apple, Adobe, Bloomberg, etc.

Dado um array de inteiros *nums* e um inteiro *target*, retorne os índices dos dois números de tal forma que eles somem *target*. Você pode assumir que cada entrada terá exatamente uma solução, e você não pode usar o mesmo elemento duas vezes. Você pode retornar a resposta em qualquer ordem.

Exemplo 1:

Entrada	Saída
<pre>{ "nums": [8,2,7,15], "target": 9 }</pre>	[1,2]

Exemplo 2:

Entrada	Saída
<pre>{ "nums": [3,2,4], "target": 6 }</pre>	[1,2]

Exemplo 2:

Entrada	Saída
<pre>{ "nums": [3,5,7,2,4,8,1,6], "target": 15 }</pre>	<pre>[2,5]</pre>

Assinaturas:

Javascript:

```
function twoSum(nums, target)
```

Java:

```
public static int[] twoSum(int[] nums, int target)
```

C#:

```
public static int[] twoSum(int[] nums, int target)
```

Python:

```
def two_sum(nums, target)
```

Problema "transacoes" (ref: *Leetcode invalid-transactions*)

Empresas: Apple, Google, Amazon, Microsoft, Adobe, Accenture, etc.

A transaction is possibly invalid if:

- the amount exceeds \$1000, or;
- if it occurs within (and including) 60 minutes of another transaction with the same name in a different city.

You are given an array of strings transaction where transactions[i] consists of comma-separated values representing the name, time (in minutes), amount, and city of the transaction. Return a list of transactions that are possibly invalid. You may return the answer in any order.

Constraints:

- transactions.length <= 1000
- Each transactions[i] takes the form "{name},{time},{amount},{city}"
- Each {name} and {city} consist of lowercase English letters, and have lengths between 1 and 10.
- Each {time} consist of digits, and represent an integer between 0 and 1000.
- Each {amount} consist of digits, and represent an integer between 0 and 2000.

Exemplo 1:

Entrada
["alice,20,800,mtv","alice,50,100,beijing"]
Saída
["alice,20,800,mtv","alice,50,100,beijing"]

Exemplo 2:

Entrada
["alice,20,800,mtv","alice,50,1200,mtv"]
Saída
["alice,50,1200,mtv"]

Exemplo 3:

Entrada
["alice,20,800,mtv","bob,50,1200,mtv"]
Saída
["bob,50,1200,mtv"]

Assinaturas:

Javascript:

```
function invalidTransactions(transactions)
```

Java:

```
public static List<String> invalidTransactions(String[] transactions)
```


C#:


```
public static List<string> InvalidTransactions(string[] transactions)
```

Python:


```
def invalid_transactions(transactions)
```



Accepted

 nelioalvesdev submitted at Jul 15, 2024 11:13

 Solution

⌚ Runtime

52 ms | Beats 100.00% 

 Analyze Complexity

💾 Memory

53.42 MB | Beats 98.53% 