

CAPÍTULO 8

Armazenamento sem repetição com busca rápida

"Não é a montanha que nos faz desanimar, mas a pedrinha que trazemos no sapato"

— Autor desconhecido

8.1 – MOTIVAÇÃO

A impressão digital é a marca deixada pelas papilas dos dedos ao tocar em algum objeto. A íris é uma parte do olho humano que pode ser usada para identificar uma pessoa. Acredita-se que ambos, impressão digital e íris, são únicos para cada indivíduo, ou seja, não há duas impressões digitais iguais ou duas íris iguais.



Figura 8.1: Impressão Digital



Figura 8.2: Íris

É importante perceber também que manter as digitais em seqüência pode não ajudar em nada. Por exemplo, suponha que a polícia coloque as digitais em ordem alfabética do nome das pessoas. Quando um policial, na cena de um crime, recolhe uma digital, ele não sabe o nome da pessoa que tem aquela digital pois se soubesse já teria resolvido o crime. Logo, o fato das digitais estarem em alguma seqüência não serve de nada para o policial.

Além de armazenarmos essas impressões digitais, uma operação chave sobre elas é poder consultar se determinada digital está contida em um grupo particular de digitais. Aqui não podemos fazer uma busca sequencial, comparando cada digital da base com a digital que está sendo procurada: isso consumiria tempo linear, o que seria inviável para nossa aplicação, que possui uma base de dados grande. Queremos evitar o consumo de tempo linear durante a busca dessa digital.

As cores são a interpretação que o cérebro faz das ondas de luz. O comprimento de uma onda de luz varia de $7.5 \times 10^{-7}m$ (vermelho) e $4 \times 10^{-7}m$ (violeta). Cada comprimento de luz é interpretado pelo cérebro como uma cor diferente e única. Em outras palavras, não há cores repetidas.

Comprimento de onda/cor identificada pelo cérebro



Figura 8.3: Cores

Os mapas são a representação diagramática da Terra em uma superfície plana, em um pedaço de papel por exemplo. Alguns mapas destacam os países e um fato interessante que ocorre é que os países não se repetem, ou seja, são únicos.

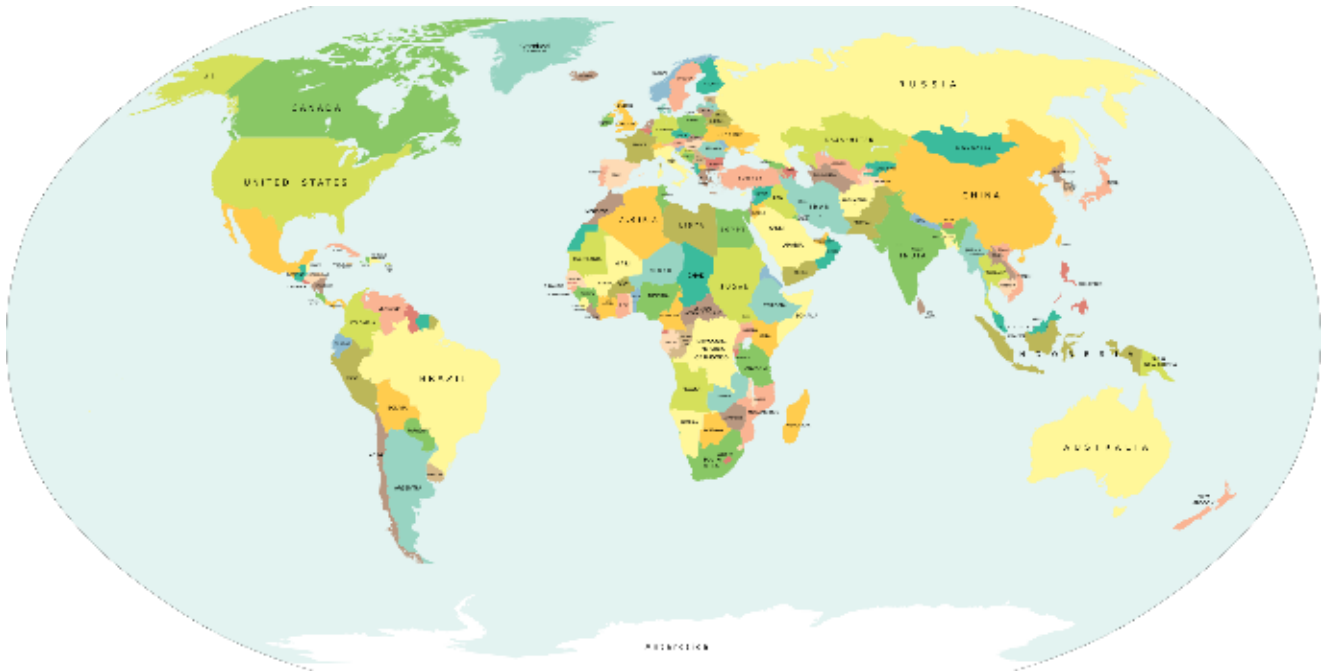


Figura 8.4: Países

São muitos os exemplos em que há a necessidade de não permitir elementos repetidos dentro de um mesmo grupo, e em especial a finalidade de poder consultar se determinado elemento se encontra neste grupo, e que esta consulta seja computacionalmente eficiente.

8.2 – O PROBLEMA DO VOCABULÁRIO

Um vocabulário é formado por um conjunto de palavras usadas na comunicação de um grupo de pessoas (um povo, uma nação, um país, etc).

Um requisito sobre as palavras que formam um vocabulário é que cada uma seja única. Não tem sentido existir palavras repetidas. Por outro lado, não há a necessidade das palavras ficarem em uma determinada seqüência. Se as palavras estiverem espalhadas sem estar numa seqüência específica, elas ainda podem formar o vocabulário.

Outro exemplo em que há a necessidade dos elementos serem únicos mas não obrigatoriamente estarem em seqüência são os baralhos. Imagine as cartas de um baralho, não pode existir cartas repetidas, elas são únicas.

Primeiro, suponha que todas as cartas estão dentro da caixa em que o baralho foi vendido. Desta forma, elas estão em alguma seqüência. Depois, suponha que todas as cartas estejam espalhadas sobre a mesa sem nenhuma seqüência. Em ambos os casos, as cartas continuam formando o baralho, independentemente se estão em seqüência ou não.

Queremos construir um sistema para armazenar as palavras de um vocabulário. Este sistema deve tomar as devidas providências para não permitir palavras repetidas.

As pessoas ao se comunicarem costumam inventar novas palavras que muitas vezes são provenientes de gírias ou inspiradas em outra língua. Algumas destas palavras, por serem tão usadas, acabam sendo incorporadas ao vocabulário de uma determinada língua. O sistema deve então fornecer uma operação para inserir palavras.

Na inserção de uma nova palavra na língua, o sistema deve verificar se a palavra já não existia pois não queremos repetir palavras. Esta verificação precisa ser rápida senão o sistema não será eficiente. Algum tipo especial de organização precisa ser usado para poder obter uma boa performance.

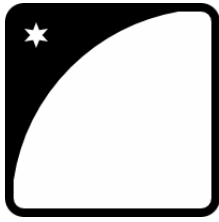
Da mesma forma que as palavras podem ser adicionadas na língua elas podem ser removidas. O sistema deve ser capaz de encontrar rapidamente as palavras que saírem do vocabulário da língua para remove-las.

Provavelmente precisaremos em alguns momentos percorrer todas as palavras armazenadas eficientemente. O sistema também deve fornecer uma funcionalidade para tal.

Outra operação útil seria alguma que informasse o número total de palavras da língua.

Você pode também fazer o curso CS-14 dessa apostila na Caelum

Querendo aprender ainda mais sobre estrutura de dados? Esclarecer dúvidas dos exercícios? Ouvir explicações detalhadas com um instrutor?



A Caelum oferece o **curso CS-14** presencial nas cidades de São Paulo, Rio de Janeiro e Brasília, além de turmas incompany.

[Consulte as vantagens do curso *Algoritmos e Estruturas de Dados com Java*.](#)

8.3 – CONJUNTOS

Vamos definir uma estrutura de dados para o sistema que armazenará o conjunto de palavras do vocabulário de alguma língua. Chamaremos esta estrutura de **Conjunto**.

Nesta seção, estamos interessados apenas em definir a interface de uso de um Conjunto. Mais adiante cuidaremos de implementar esta estrutura de dados.

As funcionalidades de um Conjunto são basicamente estas:

1. Adicionar um dado elemento.
2. Verificar se dado elemento está ou não no Conjunto.
3. Recuperar todos os elementos.
4. Informar o número de elementos.

CAPÍTULO ANTERIOR:

[Filas](#)

PRÓXIMO CAPÍTULO:

[Tabelas de Espalhamento](#)

Você encontra a Caelum também em:

[Blog Caelum](#)

[Cursos Online](#)

[Facebook](#)

[Newsletter](#)

[Casa do Código](#)

[Twitter](#)