1 → [1, 7, 9, 12]

2 → ['Palavras', 'Pedro', ...] ↺ (=)

3 → ['shingl', 'hingle', ...]

pré-processamento para ger. shingles

PL 4

Algoritmos Probabilísticos

Para analisar o numero de K-shingles basta ver quantos se repetem

Número do Grupo (N): 9

Secção para avaliação [1]

Considere uma aplicação, a desenvolver em Matlab, com algumas funcionalidades de um sistema online de apoio a viagens. A aplicação deve considerar um conjunto de turistas clientes, identificados por um ID e tendo associado um conjunto de países visitados, identificados por um ID (ambos os IDs definidos por um inteiro positivo).

Dados de entrada:

Considere o ficheiro `travelsN.data`, com N igual ao número atribuído ao seu grupo, disponível em **`http://bit.ly/483FZ2w`** e utilize os dados das duas primeiras colunas deste ficheiro para identificar os turistas e os países que cada um visitou, respetivamente. A terceira coluna contém a duração da visita (em dias). A quarta coluna do ficheiro contém a avaliação atribuída por cada turista.

A informação sobre cada um dos turistas encontra-se num segundo ficheiro, `touristsN.txt`, com conteúdo similar ao seguinte:

ID Nome Próprio Apelido interesses

```
4 ; Carol ; Jesus ; Música ; Fotografia ; Filmes ; Jogos ; Leitura ; ...
49 ; Naísa ; Rodrigues ; Fotografia ; Viagens ; Futebol ;  ...
```

em que os dados de cada coluna estão separados por ";". A linha número n contém a informação do turista com o ID n usado no ficheiro `travelsN.data`. A primeira coluna contém o número, a segunda o nome (próprio) e a terceira o apelido. As restantes colunas contêm interesses do turista, como, por exemplo, "Música".

NOTA: executando no Matlab a instrução: `dic= readcell('touristsN.txt', 'Delimiter',';');` é criado o cell array `dic` em que a célula `dic{i,j}` contém a informação da linha i e da coluna j do ficheiro.

O ficheiro `country_info.~~txt~~` (csv), com as colunas separados também por `';'` apresenta informação sobre os países. A linha número n contém a informação do país com o ID n (usado na segunda coluna do ficheiro `travelsN.data`. A primeira coluna contém o nome do país (designação comum); a segunda um pequeno texto sobre o país.

NOTA: executando no Matlab a instrução:
`countries = readcell ('countries_info.csv',Delimiter=';');`
é criado o cell array `countries` em que a célula `countries{i,j}` contém a informação da linha i e da coluna j do ficheiro `countries_info.csv`.

[1] A execução desta secção será objeto de avaliação. Assim, deverá fazer um relatório em PDF com todos os códigos Matlab desenvolvidos devidamente explicados e as opções de desenvolvimento devidamente justificadas. O relatório deverá começar por identificar o ano letivo, a disciplina, a turma prática e os elementos do grupo (nome e No. Mec.) que realizou o trabalho. **Deverá submeter um ficheiro comprimido com o relatório e todos os ficheiros necessários à execução da aplicação desenvolvida. Tenha em atenção os prazos estipulados**

Relatório pequeno, a explicar o que fizeram. Algoritmo da solução!

1 Relatar no relatório os experiências para o número de funções de hash?

Descrição da aplicação a desenvolver:

A aplicação deve começar por pedir o ID do utilizador (turista) que se torna o utilizador atual [2]:

```
Insert Tourist ID (1 to ??):
```

certificando-se que o número introduzido é um ID válido (no ficheiro `travelsN.data` os IDs dos utilizadores são números inteiros desde 1 até ao número de utilizadores distintos). Em seguida, a aplicação deve permitir ao utilizador seleccionar uma de 6 opções:

```
1 - Countries (or regions) visited by current user.
2 - Set of countries evaluated by the 2 more similar users.
3 - Suggestion of countries to visit.
4 - Suggestion of similar tourists based on interests.
5 - Estimate total of visits to the countries visited by current user.
6 - Exit
Select choice:
```

Opção 1: A aplicação lista os nomes dos países que o utilizador actual visitou. Cada linha deve mostrar, em colunas devidamente alinhadas, o ID do país e o nome.

Opção 2: A aplicação lista o conjunto dos países avaliados pelos 2 utilizadores mais "similares" ao utilizador atual. Deve ser apresentado um único conjunto, agregando os resultados.
A aplicação começa por determinar quais os utilizadores mais similares ao utilizador atual (em termos do conjunto de países visitados por cada um) e, finalmente, a aplicação lista todos os países (IDs e nomes comuns).

Opção 3: Para cada país que o utilizador atual visitou mais de 3 dias, determinar o país mais similar tendo por base o texto descritivo de cada país disponibilizado no ficheiro `countries_info.csv`. Deste processo deve resultar um conjunto de países ainda não visitados e respetivas distâncias. No final, a aplicação apresenta os IDs e os nomes dos países com distâncias inferiores à média de todas as distâncias obtidas.

Opção 4: Implementar comparação baseada na similaridade do conjunto de campos associados a cada um dos turistas (obrigatoriamente por MinHash).

Opção 5: A aplicação deve indicar uma estimativa do número de visitas (independentemente da duração, sem excluir o utilizador atual) para todos os países visitados pelo utilizador atual. A estimativa tem de ser obrigatoriamente obtida através da utilização de filtro(s) de Bloom com contagem.

Opção 5: A aplicação termina.

Notas sobre a implementação das funcionalidades da aplicação a desenvolver:

A estimativa da similaridade entre conjuntos (i.e., entre os conjuntos de turistas que visitaram cada país na Opção 2, entre 2 vetores de caracteres na Opção 3, e entre conjuntos de interesses de turistas na opção 4) tem de ser obrigatoriamente implementada por um método *MinHash*.

Na Opção 2, pode reutilizar, com as necessárias alterações, a implementação que efetuou na secção 4.3 deste guião (PL04). O número adequado de funções de dispersão k deve ser escolhido efetuando experiências com vários valores e tendo em contas as conclusões que retirou nessa altura.

Na Opção 3, deve desenvolver um método, combinando Shingles e *MinHash*, adequado a estimar a similaridade entre dois vetores de caracteres (os textos descritivos de 2 países), escolhendo de forma fundamentada tanto o tamanho dos *shingles* como o número adequado de funções de dispersão k (sugere-se que experimente tamanhos de *shingle* entre 7 e 10 caracteres).

Na Opção 4, deve desenvolver um método *MinHash* adequado à similaridade entre conjuntos de vetores de caracteres (interesses dos turistas).

[2] Para introdução de dados pelo teclado, investigue a utilidade da função Matlab `input`

MinHash só no Primeiro | O prof vai correr só o segundo 8)

Requisitos para a implementação em Matlab + Para funções !!

É obrigatório desenvolver 2 scripts Matlab que cumpram o seguinte:

O primeiro corre uma única vez para ler os ficheiros de entrada e guardar em ficheiro todas as estruturas de dados associadas aos turistas e aos países, incluindo:

- todas as matrizes de assinaturas necessárias para a resolução das várias opções;
- a(s) estrutura(s) de dados do Counting Bloom filter necessárias para todas as opções que necessitem da sua utilização

O segundo script começa por ler do disco todas as estruturas previamente guardadas pelo primeiro script e depois implementa todas as interações com o utilizador descritas anteriormente. Será fortemente penalizada o cálculo neste script de estruturas que devem ser previamente calculadas no primeiro script e/ou a não leitura da informação no início do script.

Avaliação do trabalho:

1. Opção 1 a funcionar corretamente **(máximo 2 valores)**
2. Opção 2 a funcionar corretamente **(máximo 4 valores)**
3. Opção 3 a funcionar corretamente **(máximo 4 valores)**
4. Opção 4 a funcionar corretamente **(máximo 5 valores)**
5. Opção 5 a funcionar corretamente **(máximo 2 valores)**
6. Fundamentação/avaliação das opções tomadas na implementação dos métodos probabilísticos (exemplos: número de funções de dispersão, tamanho de *shingles*, dimensionamento dos filtros de Bloom) **(máximo 2 valores**
7. Qualidade do relatório **(máximo 1)**

Boa sorte!