

Trabalho Prático

Programação 2 - BSI - Ifes Serra

22 de maio de 2024

1 Aplicativo de Fotos

Você é viciado em um famoso aplicativo de fotos, no qual você posta fotos do seu dia a dia, segue pessoas em que você possui interesse e também pode ser seguido por outras pessoas. Porém, com o tempo, você começou a questionar o algoritmo que o aplicativo usa para definir o que é relevante para você (e se você é relevante para os outros). Portanto, você decidiu implementar seu próprio aplicativo, em que as postagens de todas as pessoas que um usuário segue poderão ser ordenadas de acordo com os critérios que o próprio usuário definir (por exemplo, simplesmente classificá-las das mais recentes para as mais antigas).

A versão inicial do aplicativo armazena apenas um conjunto limitado de informações sobre cada usuário em um dicionário do Python. A chave do dicionário é o login do usuário, e o conteúdo é uma tupla com 4 informações:

1. Nome do usuário;
2. Lista de usuários que ele segue;
3. Lista de usuários que o seguem;
4. Lista de postagens que ele já fez.

Cada item da lista de postagens é uma tupla com o nome da foto no banco de dados (por exemplo, “foto1.jpg”), a legenda da foto (por exemplo, “Saudade desse dia! #tbt”) e a data da postagem. A data, por sua vez, é uma tupla com dia, mês, ano, hora e minuto da postagem, nessa ordem.

2 Objetivo

Seu trabalho considera que há um arquivo binário já existente chamado “*usuarios.bin*”, na mesma pasta onde o arquivo principal do trabalho está sendo executado, contendo apenas o dicionário com os dados de todos os usuários já cadastrados. Seu programa não vai realizar nenhum cadastro (todos os dados já virão prontos no dicionário que está neste arquivo binário).

Após a leitura dos dados, você precisa imprimir, para cada usuário, seu nome, o número de pessoas que ele segue, e o número de usuários que o seguem, como no formato a seguir:

```
Heloisa Alves (segue 5, seguido por 4)
Isabella Oliveira (segue 4, seguido por 4)
Isis Barbosa (segue 4, seguido por 4)
Valentina Rodrigues (segue 4, seguido por 4)
Theo Lopes (segue 2, seguido por 4)
```

Heitor Fernandes (segue 3, seguido por 3)
Ravi Dias (segue 3, seguido por 2)
Elisa Rodrigues (segue 2, seguido por 2)
Helena Rocha (segue 3, seguido por 1)
Pedro Oliveira (segue 3, seguido por 1)

Note que os dados estão classificados segundo os seguintes critérios de ordenação:

1. Número de seguidores (do maior para o menor);
2. Em caso de empate, pelo número de usuários que ele segue (também em ordem decrescente);
3. Em caso de novo empate, em ordem alfabética do nome do usuário;
4. Por fim, em último caso, em ordem alfabética de seus logins (não há dois usuários com o mesmo login, logo não haverá empate neste último critério).

Após a impressão dos dados acima, segundo os critérios mencionados, você irá imprimir uma linha com uma String contendo 3 hífen seguidos, para indicar que a partir dali irá iniciar a segunda saída do trabalho.

A segunda coisa a ser impressa é o feed do primeiro usuário que foi impresso acima. O feed de um usuário é o conjunto de postagens feitas por todos os usuários que ele segue. Para cada postagem, você irá exibir 5 linhas:

1. O login do usuário;
2. O nome da imagem que foi salva no banco de dados;
3. A legenda da foto;
4. A data em que a postagem foi feita;
5. Uma linha com três asteriscos, para separá-lo da próxima postagem.

As postagens deverão ser impressas no formato a seguir:

Feed de Heloisa Alves:

valentina.rodrigues

foto3.jpg

:)

26/12/2023 15:48

elisa.rodrigues

foto4.jpg

Do story pro feed

14/11/2023 09:54

isabella.oliveira

foto1.jpg

:)

12/11/2023 17:50

isabella.oliveira
foto3.jpg
Sextou!
28/10/2023 17:52

isabella.oliveira
foto0.jpg
Ta pago
14/10/2023 13:04

valentina.rodrigues
foto0.jpg
Dia lindo
03/10/2023 11:35

helena.rocha
foto2.jpg
Sextou!
13/09/2023 12:25

valentina.rodrigues
foto2.jpg
Sextou!
07/09/2023 10:09

valentina.rodrigues
foto1.jpg
Do story pro feed
17/08/2023 11:17

theo.lopes
foto3.jpg
Dia lindo
16/08/2023 15:49

theo.lopes
foto2.jpg
Biscoito
25/07/2023 08:45

helena.rocha
foto3.jpg
Dia lindo
22/07/2023 08:02

elisa.rodrigues
foto1.jpg
:)
07/07/2023 17:24

elisa.rodrigues
foto2.jpg
Ta pago
12/06/2023 18:47

elisa.rodrigues
foto3.jpg
:)
24/05/2023 10:43

helena.rocha
foto1.jpg
Sextou!
11/05/2023 09:12

isabella.oliveira
foto2.jpg
Ta pago
27/04/2023 19:25

theo.lopes
foto0.jpg
Biscoito
24/04/2023 21:34

isabella.oliveira
foto4.jpg
TBT
10/04/2023 11:22

elisa.rodrigues
foto0.jpg
Do story pro feed
19/03/2023 17:56

theo.lopes
foto1.jpg
Dia lindo
19/02/2023 15:34

helena.rocha
foto0.jpg
Sextou!
18/01/2023 15:36

helena.rocha
foto4.jpg
Do story pro feed
18/01/2023 09:33

```
***
valentina.rodrigues
foto4.jpg
Ta pago
15/01/2023 15:37
***
```

Note que, dessa vez, os critérios de classificação das postagens são:

1. A data de postagem (da mais recente para a mais antiga);
2. Em caso de empate no dia, mês, ano, hora e minuto, as postagens também serão classificadas em ordem alfabética do login do usuário.

Dessa forma, nenhuma pessoa que você segue será mais “relevante” que outra em seu feed. Você terá certeza que vai visualizar todas as postagens, sempre ordenadas da mais recente para a mais antiga.

3 Restrições

Como os dados estão dispostos em um dicionário, e não é possível ordenar um dicionário, na primeira parte do programa você deverá criar uma lista contendo **apenas** os logins dos usuários. Apenas esta lista deverá ser ordenada, de acordo com os critérios mencionados acima. Nenhuma outra informação além do login pode ser salva nesta lista, nem em nenhuma outra lista auxiliar.

Na segunda parte, a estrutura é livre. Você pode, por exemplo, criar uma nova lista representando o feed, contendo as 4 informações necessárias (login, foto, legenda e data) de cada postagem de cada pessoa que o usuário segue, e depois ordenar essa lista pelos critérios mencionados.

Cada ordenação deve ser feita de uma única vez (por exemplo: para ordenar a lista de logins na primeira parte, você chama uma única função de ordenação que vai classificar a lista de uma única vez segundos **todos** os critérios necessários). Além disso, é obrigatório que o método de ordenação utilizado tenha uma complexidade de $O(n \times \lg n)$ no caso médio para uma lista de n elementos.

Por fim, **nada será impresso na tela**. Toda a saída será impressa num arquivo texto chamado “*saida.txt*”, no formato idêntico ao especificado aqui. Há arquivos de exemplo na página da disciplina para você comparar sua saída com a saída correta (há arquivos de entrada e saída para 10, 100, 1.000, 10.000 e 100.000 usuários).

4 Observações

1. Este trabalho vale 40 pontos e deve ser entregue até 06/06.
2. Os trabalhos que executarem mais rápido terão 2 pontos de bônus (desde que salvem um arquivo de saída com a resposta correta).
3. Deve ser enviado apenas um único arquivo .py (ou um arquivo compactado contendo apenas os *.py, caso haja mais de um). Não envie nenhum arquivo de entrada/saída, apenas os códigos fonte.
4. A atividade pode ser feita em grupos de até **dois** integrantes.

5. Atividades entregues após o prazo, com erro de execução, com formato de saída incorreto ou que não compilarem terão nota 0.
6. Atividades consideradas plágio terão nota 0 para quem copiou e para quem forneceu a atividade, e serão enviadas ao Conselho de Ética.
7. A atividade deve ser enviada na sala da disciplina do AVA.
8. Em caso de dúvidas na especificação da atividade ou na própria atividade, contate-me em sala de aula.