

LAB 02

MC558A

Data de Entrega: 01/09

Fomos contratados para construir um bot que vai visitar diversas redes sociais, fóruns e sites de compras, para coletar as atividades online dos usuários. Nosso objetivo final é construir uma ferramenta capaz de sugerir propagandas e serviços mais alinhados ao perfil de cada pessoa com base no seu comportamento online.

Porém, estamos enfrentando um grave problema. Cada pessoa pode ter um nome de usuário diferente em cada site que usa, e mesmo usuários com o mesmo nome, em sites diferentes podem não ser a mesma pessoa, assim, precisamos descobrir como identificar os usuários e relacionar as informações coletadas dessas diversas fontes.

Já conseguimos extrair de cada site o apelido de cada usuário e os emails que cada usuário já teve cadastrado naquele site. Queremos que através desses emails você relacione os usuários!

Por exemplo, analisando 4 sites, encontramos os seguintes usuários e emails:

Site 1:

Ana-dias : ana-d@tmail.com, ana.dias@yahuu.com

ONinja : rui@coldmail.com, oNinja@tmail.com

Site 2:

Rui42 : rui.r@yahuu.com, rui@coldmail.com

Bbia : bia@coldmail.com, bb@coldmail.com

Site 3:

Rui.Dias40 : rui.r@yahuu.com

PedroA : pedro@yahuu.com, pedrinho@coldmail.com

Gil.P : gil.p@tmail.com

Site 4:

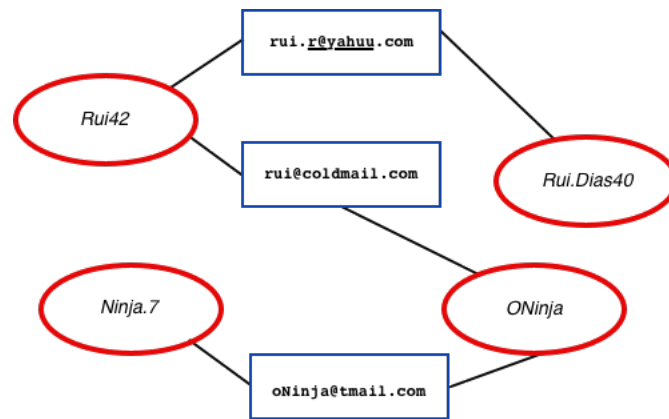
Ana97 : ana.dias@yahuu.com.

Rui.Silva : rui.silva@coldmail.com

Ninja.7 : oNinja@tmail.com.

Através dos emails podemos identificar os usuários que são a mesma pessoa.

Por exemplo, o email *ana.dias@yahuu.com*, é comum ao usuário “*Ana-dias*” do site 1 e também do usuário “*Ana97*” do site 4, logo, vamos assumir que são a mesma pessoa.



O email *rui.r@yahoo.com* é encontrado no usuário “*Rui42*” do site 2 e também no usuário “*Rui.Dias40*” do site 3. Aqui já temos algo mais complexo, veja que o segundo email que esse usuário já teve é *rui@coldmail.com*, encontramos também esse email com o usuário “*ONinja*” do site 1, e esse mesmo usuário “*ONinja*” do site 1, tem outro novo email cadastrado, “*oNinja@tmail.com*”, que é também encontrado com o usuário “*Ninja.7*” do site 4. Portanto, *Rui42*, *Rui.Dias40*, *ONinja* e *Ninja.7* estão relacionados, e também vamos assumir que pertencem a mesma pessoa!

Resumindo, você precisa agrupar todos os usuários que pertencem ao mesmo componente de emails. A saída terá que informar quantas pessoas foram encontradas e quais apelidos de usuários pertencem a mesma pessoa.

Nosso time de backend já vai pré-processar algumas informações, precisamos que você foque somente no componente que irá fazer a análise. Lembre-se também que vamos utilizar seu algoritmo em grandes volumes de dados, precisamos de algo bem otimizado!

Formato de entrada

N M
 user1
 user2
 ...
 userN
 email1
 email2
 ...
 emailM
 $p_1 \ e_i^1 \ e_h^2 \dots \ e_k^{p_1}$
 $p_2 \ e_z^1 \ e_n^2 \dots \ e_i^{p_2}$

Formato de saída

K
 user1, user2, user14... user70
 user3, user4, user7
 user5, user8
 ...

Descrição da entrada : Dois inteiros N e M representando o número de usuários e o número de emails, seguidos de N linhas contendo os nomes dos usuários, e M linhas contendo os emails. Por fim, N linhas contendo o relacionamento entre usuários e emails. Cada linha é iniciada por um inteiro p que indica o número de emails que o n -ésimo usuário tem relacionado e o id dos emails relacionados.

Todo id é um número inteiro entre 0 e 100.000

Descrição da saída : Um inteiro K, representando o número de pessoas encontradas, seguidas de K linhas informando por linha o nome dos usuários que são da mesma pessoa.
IMPORTANTE: A saída deve estar ordenada em relação ao identificador dos usuários!

Exemplo 1 :

Exemplo de entrada

```
10 11
Ana-dias
ONinja
Rui42
Bbia
Rui.Dias40
PedroA
Gil.P
Ana97
Rui.Silva
Ninja.7
ana-d@tmail.com
ana.dias@yahuu.com
rui@coldmail.com
oNinja@tmail.com
rui.r@yahuu.com
bia@coldmail.com
bb@coldmail.com
pedro@yahuu.com
pedrinho@coldmail.com
gil.p@tmail.com
rui.silva@coldmail.com
2 0 1
2 2 3
2 4 2
2 5 6
1 4
2 7 8
1 9
1 1
1 10
1 3
```

Exemplo de saída

```
6
Ana-dias Ana97
ONinja Rui42 Rui.Dias40 Ninja.7
Bbia
PedroA
Gil.P
Rui.Silva
```

Analisando o Exemplo 1: 10 é o numero de usuários e 11 o número de emails.

Ana-dias, ONinja, (...), Ninja.7 são os apelidos dos usuários. ana-d@tmail.com, ana.dias@yahuu.com, rui@coldmail.com, oNinja@tmail.com (...), rui.silva@coldmail.com são os emails. Ana-dias (o primeiro usuário), tem 2 emails registrados: ana-d@tmail.com e ana.dias@yahuu.com (emails 0 e 1). O usuário ONinja (segundo usuário) tem e os emails registrados : rui@coldmail.com, oNinja@tmail.com (emails 2 e 3). Assim até o ultimo usuário, *Ninja.7*, com 1 email registrado : oNinja@tmail.com (email 3).

Observações e Avaliação:

- É permitido SOMENTE o uso das estruturas *vector*, *queue*, *map* e *stack*. Nenhuma outra estrutura de dados sofisticada poderá ser utilizada das bibliotecas do C/C++. Caso julgue necessário o uso de alguma outra estrutura, implemente.
- É esperado do seu código complexidade de tempo linear em relação ao tamanho da entrada. Códigos com complexidades piores terão a nota reduzida.
- Os programas que não estiverem compilando ou não passarem em algum dos testes pré-instalados no SuSy, terão nota 0.
- O arquivo fonte deve estar bem comentado! Qualquer função ou trecho de código não trivial deve conter uma breve descrição sobre o seu propósito.
- No início do arquivo fonte enviado ao SUSY, deverá haver uma descrição em alto nível sobre como a solução foi pensada e implementada. Também é necessário justificar porque o algoritmo tem complexidade linear. É esperado uma descrição com cerca de 100 a 200 palavras.
- Um esqueleto da aplicação está disponível na pagina da disciplina, o uso é obrigatório. Entrada e saída de dados já estão disponíveis na main. É necessário somente implementar a função *solve()*, você pode implementar também outras funções auxiliares caso julgue necessário. Para compilar, basta executar o comando *make* no endereço do fonte, para testar com o exemplo, basta executar *make run*.
- É esperado que esta atividade demande cerca de 2 a 6 horas para ser finalizada, caso esteja demorando muito mais tempo, ou caso tenha qualquer dúvida, procure atendimento o quanto antes!
- Em caso de plagio, todos os alunos envolvidos serão imediatamente reprovados.