Desenvolvedor (a) de Busca - Desafio

Bem vindo ao desafio técnico do Luizalabs e muito obrigado por se interessar em

fazer parte do Magazine Luiza!

O objetivo deste desafio é avaliar seu conhecimento sobre algoritmos de indexação e de recuperação de informação, avaliar a qualidade do seu código em termos de

escrita, padronização e performance na utilização de recursos e estruturas de dados.

**O Problema**

Estamos disponibilizando para você, no mesmo e-mail que este desafio chegou, o arquivo movies.zip que possui arquivos contendo dados de filmes.

Seu desafio é criar um programa que busque por uma sentença em todos estes

arquivos e exiba quantos e quais arquivos possuem esta palavra chave. A ordenação

dos arquivos deve ser feito em ordem crescente e alfabética. O seu programa deve ser executado a partir da linha de comando do terminal (Linux/MacOS) e deve permitir que a sentença que deverá ser encontrada seja passada como parâmetro de execução.

Exemplo de comando de execução com a sentença "walt disney" sendo passada para o programa:

python search.py “walt disney”

ou

java -jar search.jar “walt disney”

Exemplo de resposta do programa:

Foram encontradas 53 ocorrências pelo termo "walt disney".

Os arquivos que possuem "walt disney" são:

data/a-cowboy-needs-a-horse.txt

data/alice-and-the-three-bears.txt

data/alice-helps-the-romance.txt

data/alice-s-fishy-story.txt

data/alpine-climbers.txt

data/billposters.txt

data/bone-trouble.txt

data/bootle-beetle.txt

data/canine-caddy.txt

data/clown-of-the-jungle.txt

data/cock-o-the-walk.txt

data/cured-duck.txt

data/dog-watch.txt

data/donald-s-dream-voice.txt

data/donald-s-snow-fight.txt

data/dude-duck.txt

data/el-gaucho-goofy.txt

data/fall-out-fall-in.txt

data/father-noah-s-ark.txt

data/fathers-are-people.txt

data/football-now-and-then.txt

data/funny-little-bunnies.txt

data/gallopin-gaucho.txt

data/good-scouts.txt

data/goofy-and-wilbur.txt

data/great-guns.txt

data/how-to-be-a-sailor.txt

data/how-to-have-an-accident-at-work.txt

data/jiminy-cricket-s-christmas.txt

data/mickey-s-elephant.txt

data/moochie-of-pop-warner-football.txt

data/on-ice.txt

data/pantry-pirate.txt

data/perri.txt

data/self-control.txt

data/sky-scrappers.txt

data/society-dog-show.txt

data/spare-the-rod.txt

data/tall-timber.txt

data/the-four-musicians-of-bremen.txt

data/the-hockey-champ.txt

data/the-jazz-fool.txt

data/the-little-house.txt

data/the-mail-pilot.txt

data/the-merry-dwarfs.txt

data/the-nifty-nineties.txt

data/the-pet-store.txt

data/tommy-tucker-s-tooth.txt

data/tugboat-mickey.txtdata/two-gun-mickey.txt

data/undiscovered-walt-disney-world.txt

data/wide-open-spaces.txt

data/working-for-peanuts.txt

**Requisitos**

* Você deve ter o código coberto por testes unitários e deve utilizar somente ferramentas nativas da linguagem que você escolher no desenvolvimento do algoritmo (Node.js, Java ou outra de sua preferência).
* É desejável que o código esteja versionado, preferencialmente usando GIT. Para os testes você pode utilizar frameworks não nativos
* Seu projeto deve ser entregue com um arquivo README.md que contenha as instruções de instalação, execução de testes e execução do programa.
* Lembre-se que as palavras digitadas em um sistema de busca não necessariamente estão na mesma ordem em que elas estão salvas nos arquivos.
* O "match" das palavras deve utilizar o critério "AND". Ou seja, ambas devem existir no arquivo, independente da ordem ou das outras palavras separarem múltiplos termos.
* O processo de busca deve ser executado em até 1 ms.
* Você pode pré-processar e preparar a massa de arquivos em um outro script

**Considerações finais**

Evite o uso de frameworks que criem muitas abstrações, queremos ver como você desenvolve e estrutura o projeto.

Será julgado seu poder de síntese, de fazer mais com menos, de arquitetar a solução da forma mais simples e robusta possível, de uma forma que seja manutenível, com qualidade e elegante.

Você pode, quando tiver qualquer tipo de dúvida sobre o desafio, nos encaminhar um email com essas perguntas.

Esperamos que você consiga concluir o desafio em 1 semana, mas se precisar de mais tempo nos informe.

Esperamos que se divirta no processo,

Boa sorte!