

TRABALHO PRÁTICO PARTE 4

PRAZO PARA ENTREGA: 20/02/2021

1. Introdução

O TikTok é um aplicativo bastante popular que permite aos usuários compartilhar vídeos curtos. O aplicativo para Android pode ser baixado pela Google Play. A base de dados disponível em <https://www.kaggle.com/shivamb/35-million-tiktok-mobile-app-reviews> [1] reúne informações de aproximadamente 3,5 milhões de *reviews* de usuários do aplicativo na loja da Google. Esses dados podem ser encontrados no link supracitado, no arquivo *tiktok_app_reviews.csv*. Cada registro do arquivo é composto das seguintes informações:

- `review_id`: string identificadora única
- `review_text`: o *review* postado pelo usuário
- `upvotes`: número de votos favoráveis ao *review*
- `app_version`: string indicando a versão do aplicativo a que o *review* se refere
- `posted_date`: data e hora de postagem

2. Objetivos

O objetivo principal deste trabalho é dar ao aluno a oportunidade de analisar e comparar o desempenho de diferentes algoritmos de ordenação aplicados a um conjunto de dados reais. Espera-se que o aluno seja capaz de:

- Manipular adequadamente arquivos em modo texto e binário.
- Compreender e implementar corretamente algoritmos de compressão;
- Analisar o desempenho dos algoritmos implementados segundo o referencial teórico adquirido na disciplina;
- Apresentar os resultados obtidos com o trabalho de maneira formal.

3. Desenvolvimento

As etapas para o desenvolvimento do trabalho são descritas abaixo.

Etapas 0: Carregamento dos dados

Nesta etapa, deve-se utilizar a implementação feita na entrega anterior para recuperar N registros aleatórios dos arquivos.

Etapa 1: Implementação do método de compressão

Implementar o método de compressão de Huffman para o processamento de uma cadeia de caracteres aleatória. Esta cadeia de caracteres será formada pela concatenação dos `review_texts` dos N registros aleatórios importados. Deve-se implementar as rotinas de compressão e descompressão.

Etapa 2: Interface com o Usuário

O programa deve possuir opções de menu para:

- Comprimir um conjunto de N (definido pelo usuário) `review_texts` aleatórios e salvá-los em um arquivo `reviewsComp.bin`;
- Descomprimir o arquivo `reviewsComp.bin` e salvar o texto resultante em um arquivo `reviewsOrig.bin`;
- Executar uma sequência de compressões de N (usar os valores 10.000, 100.000 e 1.000.000) `review_texts` aleatórios e computar as métricas indicadas na Seção 4.

4. Análise

Para gerar as estatísticas de desempenho, você deverá executar a sequência de compressões, minimamente, $M = 3$ vezes. Ao final, compute as médias de cada uma das métricas (comparações e taxa de compressão) obtidas para as M execuções. Salve todos os resultados obtidos em um arquivo `saida.txt`, contendo tanto os resultados individuais quanto a média final.

5. Relatório parcial

Você deverá confeccionar um relatório simples de no máximo uma página sobre o trabalho desenvolvido. Este relatório deve conter, obrigatoriamente, os seguintes itens:

- detalhamento das atividades realizadas por cada membro do grupo;
- decisões de implementação;
- toda e qualquer referência utilizada no desenvolvimento do trabalho.

Note que esse relatório, embora simples, deve ser formal e bem redigido. A divisão de tarefas do trabalho se aplica somente à implementação. Todo o grupo é responsável pelo relatório (escrita e revisão).

6. Exigências

O trabalho deverá, obrigatoriamente, atender aos seguintes requisitos:

- Implementação em C ou C++

- O projeto deve ser compilável e executável via linha de comando. Não conte com a presença de IDEs como Code::Blocks ou Visual Studio. Caso seu grupo opte por utilizar algum ambiente de desenvolvimento, certifique-se de que o projeto enviado possa também ser facilmente compilado em um sistema operacional Linux sem esses ambientes instalados. Forneça instruções claras e precisas de compilação e execução pela linha de comando. **Recomenda-se a utilização de algum Makefile ou script para a compilação.** Caso o grupo julgue necessário, é possível solicitar que o professor verifique as instruções de compilação **antes** do prazo final de envio.
- **Nenhum trabalho será aceito após o prazo.**
- **O programa desenvolvido deve permitir que o usuário entre com o caminho do diretório que contém os arquivos de dados como um argumento na linha de comando.** Veja o Exemplo:

```
$ ./programa /diretorio/contendo/arquivos/entrada
```

- O programa deve procurar pela existência do arquivo binário dentro da pasta (gerado na Parte 1 do trabalho). Se não existir, deve gerá-lo novamente.
- Não é permitida a utilização de bibliotecas externas. Todos os algoritmos devem ser implementados pelo grupo. Caso haja dúvidas quanto à utilização de alguma função ou biblioteca, entre em contato com o professor.
- Obviamente, todo código deve ser de autoria do grupo. Não é permitida a utilização de códigos de terceiros ou de outros grupos. É permitida a pesquisa por estratégias para a solução dos problemas (e as referências utilizadas nessas pesquisas devem constar do relatório), porém a apropriação de código alheio não será aceita. **Qualquer tentativa de plágio identificada resultará em nota zero. Os códigos fontes serão analisados pelo sistema Moss (<http://theory.stanford.edu/~aiken/moss/>)**

7. Entrega

O grupo deverá ser formado por **no máximo** 4 alunos, e as responsabilidades de cada aluno devem ser documentadas e registradas. Não é permitido que algum integrante do grupo fique responsável somente pela confecção do relatório. Todos os integrantes devem contribuir com a implementação. A distribuição das responsabilidades deve ser feita de maneira uniforme, de modo que cada membro do grupo se envolva com o trabalho na mesma proporção que os demais.

Todos os itens abaixo devem ser entregues:

1. Código-fonte completo;
 - a. Deve ser submetido um link para um repositório git (github) contendo o código do trabalho;

- b. **Não incluir as sequências e padrões na submissão via Google Classroom.**
2. Relatório em Google Doc ou PDF atendendo ao especificado na Seção 4 deste documento.

7. Critérios de avaliação

O grupo será avaliado de acordo com os seguintes critérios:

- Execução correta do código (E);
- Atendimento ao que foi solicitado (A) (valor entre 0 e 1);
- Organização do código (O): seu código deve estar bem modularizado e bem documentado;
- Qualidade do relatório apresentado (R).

Cada membro do grupo será avaliado individualmente, tanto com relação aos detalhes de implementação que ficaram sob sua responsabilidade, quanto ao entendimento em alto nível de abstração do que foi feito pelo grupo como um todo. O entendimento teórico do conteúdo relacionado ao trabalho também será avaliado. A nota individual (M) será um valor de 0 a 1 que irá ponderar a nota da implementação a ser verificada na entrevista.

A nota final de cada integrante será computada de acordo com a seguinte fórmula:

$$\text{Nota} = 0.5 * [(0.7 * E + 0.3 * O) * M * A] + 0.5 * R * A$$

O critério (A) será considerado de acordo com a composição do grupo. O objetivo é evitar que algum aluno seja prejudicado pela desistência de outros membros do grupo. **Caso algum membro do seu grupo tranque ou abandone a disciplina, comunique o professor o quanto antes para que se possa discutir alternativas.**