TRABALHO PRÁTICO
PARTE 4

PRAZO PARA ENTREGA: 09/09/2021

1. Introdução

O objetivo principal deste trabalho é dar ao aluno a oportunidade de analisar e comparar o desempenho de diferentes estruturas de dados e seus algoritmos aplicados a um conjunto de dados reais. Para tanto, será considerado como exemplo, problemas reais de encontrar padrões em sequências de DNA e, de armazenar tais sequências de forma mais eficiente. Utilize sequências geradas aleatoriamente disponíveis em:

- https://www.bioinformatics.org/sms2/random_dna.html

Espera-se que o aluno seja capaz de:

- Entender os principais algoritmos de processamento e codificação de cadeias de caracteres. Especificamente,
  - Aplicar e analisar o desempenho dos algoritmos de casamento de padrões
  - Aplicar algoritmos de compressão em cadeias de caracteres.
- Apresentar os resultados obtidos com o trabalho de maneira formal.

2. Desenvolvimento

As etapas para o desenvolvimento do trabalho são descritas abaixo.

2.1 Implementação dos métodos de Casamento de Padrão

Você deverá implementar os algoritmos vistos em aula e um terceiro método a escolha:

- Força-bruta;
- KMP;
- < terceiro método a escolha >

Para todos os algoritmos, você deve gerar uma sequência (T) aleatória de DNA com tamanho 10,000,000 e N padrões (P) de tamanho M. Os algoritmos devem retornar o número de ocorrências do padrão (P) e suas localizações em T.

2.2 Implementação do Método de Compressão

Implementar o método de compressão de Huffman para o processamento de uma sequência aleatória de DNA. Deve-se ser implementado as rotinas de compressão e descompressão.

2.3 Interface com o Usuário

Os arquivos com a sequência de DNA devem ser nomeados *dnaX.txt* onde *X* é um valor entre 1 a S e S é quantidade de sequências que serão processadas. Para os padrões, os arquivos devem ser nomeados como *padraoX.txt* onde *X* é um valor entre 1 a N.

O programa deve receber como argumentos de entrada os caminhos para a sequência e o padrão a serem utilizados. A seguir, deve oferecer um menu para realizar casamento de padrão (usar um submenu para a escolha do método) ou a compressão para atender as análises solicitadas a seguir. Caso a compressão seja selecionada, o programa deverá comprimir a sequência de DNA, calcular a taxa de compressão e descomprimir a sequência. Para qualquer método selecionado, deve-se imprimir o custo computacional em segundos.

3. Analise

Desempenho dos métodos de casamento de padrão.

Obter os tempos de processamento de cada método implementado. Para tanto, você deve executá-los para uma mesma sequência T e padrão P. Diante disso, faça:

- Tabela com diferentes tamanhos (M) de padrões aleatórios. Escolha o número de padrões (N) no mínimo igual a 5.

Taxa de Compressão

Obter as taxas de compressão para três tamanhos distintos de sequências de DNA aleatórias, definidos abaixo:

- 10.000.000
- 1.000.000
- 100.000

Além disso, deve-se calcular o custo computacional para realizar a compressão e a descompressão da sequência.

4. Relatório

Você deverá confeccionar um relatório detalhado sobre o trabalho desenvolvido. Este relatório deve conter, obrigatoriamente, os seguintes itens:

- Detalhamento das atividades realizadas por cada membro do grupo;
- Explicações dos algoritmos escolhidos, bem como as razões para cada escolha;
- Apresentação dos resultados através de tabelas e gráficos que permitam visualizar com clareza o que foi obtido;
- Análise detalhada dos resultados obtidos;
- Toda e qualquer referência utilizada no desenvolvimento do trabalho.

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

5. 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, Visual Studio ou NetBeans. 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.
- **Trabalho entregue após o prazo será penalizado.**
- Não é permitida a utilização de bibliotecas externas. As estruturas de dados e seus algoritmos devem ser implementados pelo grupo. Um dos objetivos do trabalho é que vocês aprendam a trabalhar com os algoritmos escolhidos. Se você tiver dúvida 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/)**

6. 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.**