

Trabalho T1

- Este trabalho consiste no desenho, análise e implementação de algoritmos greedy e de divisão e conquista.
- O trabalho pode ser realizado grupos de até 5 alunos.
- O trabalho deve ser implementado em Java.

Objetivos do trabalho:**Problema 1**

Alguns de seus amigos entraram no crescente campo da mineração de dados de séries temporais, onde procuramos padrões em sequências de eventos que ocorrem ao longo do tempo. As ordens de compra nas bolsas de valores são uma fonte de dados com uma ordenação natural no tempo. Dada uma longa sequência S de tais eventos, seus amigos querem uma maneira eficiente de detectar certos “padrões” neles – por exemplo, eles podem querer saber se os quatro eventos

$$S = \{\text{“buy Google”, “buy Apple”, “buy Google”, “buy NVIDIA”}\}$$

ocorrem nesta sequência S , em ordem, mas não necessariamente consecutivamente.

Eles começam com uma coleção de eventos possíveis (por exemplo, as transações possíveis) e uma sequência S de n desses eventos. Um determinado evento pode ocorrer múltiplas vezes em S (por exemplo, as ações do Google podem ser compradas muitas vezes em uma única sequência S). Diremos que uma sequência S' é uma subsequência de S se houver uma maneira de deletar certos eventos de S de modo que os eventos restantes, em ordem, sejam iguais à sequência S' . Assim, por exemplo, a sequência de quatro eventos acima é uma subsequência da sequência

$$S = \{\text{“buy Amazon”, “buy Google”, “buy Apple”, “buy Google”, “buy Google”, “buy NVIDIA”}\}$$

$$S' = \{\text{“buy Google”, “buy Apple”, “buy Google”, “buy NVIDIA”}\}$$

O objetivo deles é ser capaz de imaginar sequências curtas e detectar rapidamente se são subsequências de S . Portanto, forneça um algoritmo que receba duas sequências de eventos S' de comprimento m e S de comprimento n , cada uma possivelmente contendo um evento mais de uma vez, e decide se S' é uma subsequência de S .

Estruture a resposta no formato de um relatório com as seguintes sessões:

1. O Problema;
2. O Algoritmo;
3. Análise do Algoritmo;
4. Implementação e Tempo de Execução. O método implementado retorna verdadeiro se S_line (S') é uma subsequência de S e deve respeitar a seguinte assinatura:

boolean hasTrend(String[] S, String[] S_line)

Problema 2

Até 1969 matemáticos acreditavam ser impossível resolver um problema de multiplicação de matrizes em tempo menor que $\Theta(n^3)$. Procure pelos trabalhos de V. Strassen e responda se isso ainda é verdade.

Estruture a resposta no formato de um relatório com as seguintes sessões:

1. O Problema;
2. O Algoritmo;
3. Análise do Algoritmo;
5. Implementação e Tempo de Execução. O método implementado retorna a matriz resultante das matrizes A e B passadas por parâmetro e deve respeitar a seguinte assinatura:

int[][] multiply(int[][] A, int[][] B)

Entregáveis do Trabalho:

1. Relatório em Word com:
 - a. Capa com título e nome dos integrantes;
 - b. Resolução do problema 1;
 - c. Resolução do problema 2;
2. Código fonte comentado.

Critérios de Avaliação:

- Código fonte compilável sem erros;
- Qualidade e profundidade das análises do relatório;
- Qualidade e documentação do código fonte;
- **Respeite as assinaturas solicitadas no trabalho. Implementações diferentes da assinatura especificada não serão avaliadas, recebendo nota zero.**