

Exercício 01

Projeto e Análise de Algoritmos

Aluno: Thales Eduardo Adair Menato - 407976

Problema B

Descrição geral

O código fonte é composto de 3 arquivos: *Disciplina.java*, *Main.java*, *problemaB.java*.

Em **Disciplina** temos as definições para a classe *Disciplina* que é utilizada para conter os dados que são importados dos arquivos lidos. Dessa forma é mais fácil trabalhar com os mesmos, sendo que há *getters* e *setters* para todos os atributos dos objetos.

Em **problemaB** estão contidos métodos estáticos que são utilizados em *Main* para que possamos obter o maior índice CM desejado. Há diversas funções para escrita no console que foram utilizadas para teste, junto com debug. Temos também *lerNumOfertas* que recebe a leitura da primeira linha do arquivo, logo em seguida é utilizada a *lerDisciplina* que recebe as linhas restantes e insere cada objeto *Disciplina* dentro de um array de lista encadeada que é organizada pelo seu índice, sendo que cada índice é um dia da semana ("seg" = 0, "ter" = 1, ... , "dom" = 6), dessa forma fica mais fácil trabalhar com cada dia separadamente e os dias não existentes estão vazios.

A principal função é a *maiorCMAcumulado* que irá utilizar um loop para retornar ao array de listas encadeadas *disciplinasSelecioneadas* o maior CM de cada dia, através da função *maiorCMDia*. Dentro de *maiorCMDia* há uma recursão que não está funcionando corretamente, para casos onde há apenas 2 disciplinas para cada dia não há erro mas quando a recursão ocorre as condições para verificação não estão corretas, ele acaba retornando disciplinas onde há colisão de horários.

No **Main** é onde está localizada a função *main* onde é feito a leitura dos arquivos contidos no package *data*. E no final do método está sendo exibido no console o *problema.maiorCMAcumulado()*.

Desafios e decisões de estratégia

O código inicial estava sendo feito utilizando *loops*, foi possível realizar um código funcional para o primeiro problema mas quando foi testado o segundo, onde há 3 dias e pegar o 2º e 3º é uma melhor opção que o 1º, o código não conseguia retornar os valores corretos. Foi então que decidi mudar a estratégia e trabalhar com um código que organiza os dias separadamente através de um array de listas encadeadas, onde cada índice seria um dia da semana.

O código acabou crescendo e eu acabei me perdendo com diversos índices e erros, então decidi organizar melhor a estrutura e trabalhar com *backtrack* recursivo, a maior dificuldade é saber onde gerar a recursão e qual a condição de parada.

Por falta de experiência em funções recursivas, foi necessário um tempo imenso desenhando e analisando os casos, mesmo assim a função permanece com erro e por falta de tempo não será possível realizar a correção.